

# Analysis of Resource Transfers in Peer-to-Peer File Sharing Applications using Fluid Models

R. Gaeta, M. Gribaudo, D. Manini, and M. Sereno  
Dipartimento di Informatica, Università di Torino,  
Corso Svizzera, 185, 1049 Torino, Italia  
Phone: +39 011 670 67 70, Fax: +39 011 75 16 03  
{rossano,marcog,manini,matteo}@di.unito.it

## Abstract

This paper proposes a stochastic fluid flow model to compute the transfer time distribution of resources in peer-to-peer file sharing applications. The amount of bytes transferred among peers is represented by a continuous quantity (the fluid level) whose flow rate is modulated by a set of discrete states representing the concurrent upload and download operations on the peers participating to the transfer. A transient solution of the model is then performed to compute the probability that a peer can download a given resource in less than  $t$  units of time as a function of several system parameters. In particular, the impact of file popularity, bandwidth characteristics, concurrent downloads and uploads, cooperation level among peers, and user behavior are included in our model specification.

We also provide numerical results aiming at proving the potentialities of the approach we adopted as well as to investigate interesting issues related to the effect of incentive mechanisms on the user cooperation.

**Keywords:** peer-to-peer, file-sharing, stochastic fluid models, analytical model, transfer time distribution.

## 1 Introduction

Peer-to-peer (P2P) networking has recently emerged as a new paradigm for building network applications. In particular, in the last few years, P2P file sharing applications are generating an increasing fraction of the traffic on today's Internet and are reshaping the way new network applications are designed. The P2P approach differs from the traditional client/server approach towards building networked applications in several ways. Perhaps most importantly, the participating hosts play dual roles as servers and clients. A peer thus generates workload

for the P2P application, while also providing the capacity to process the workload generated by other peers.

P2P file sharing applications first become prominent with the introduction of Napster, which allowed users to share MP3 music files among an arbitrary set of users. Napster uses a centralized server to maintain meta information, i.e., which machine (peer) has what resources and thereby assists in locating the required resources. The actual transfer of resources, is however, done directly between the peers.

Gnutella offers a radically different approach [13]. It is fully distributed with no single point of failure. Each node has an index of its own files. File discovery is performed by flooding the network with request packets. Other P2P applications like Chord [5], CAN (Content Addressable Network) [22] and Pastry [26] are also decentralized networks with different routing schemes.

According to the SD-NAP traces [1], the dominant traffic observed by Internet Service Providers (ISPs) is associated with P2P file sharing applications. Perhaps driven by the growth in broadband services, e.g., cable and ADSL, the average resource size exchanged on P2P networks is growing, e.g., enabling the sharing of video files. Thus it is reasonable to expect the predominance of P2P traffic on the Internet to grow further.

In general, in P2P file sharing applications, the peer acting as a server (that we call *s\_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 (that we call *c\_peer*), alternates between the search of a resource and the transfer of the resource (downloading) from a *s\_peer*. Following a successful search, P2P file sharing applications provide the *c\_peer* a list of peers holding a copy of the requested resource: depending on the particular application, additional information describing the peers and the copy of the resource are included such as bandwidth between the *s\_peer* and its Internet Service Provider (ISP), number of *c\_peers* that are using this *s\_peer*, the quality of the resource, and other information the *c\_peer* may use for its *s\_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 file sharing application such as signaling, routing, searching protocols. On the contrary, the transfer time of a resource is mainly dominated by network characteristics (last and first mile bandwidth<sup>1</sup>, latency along the path connecting peers), application characteristics (the maximum number of allowed concurrent downloads and uploads, cooperation incentive mechanisms), resource characteristics (size, quality, and popularity of the resource to be downloaded, the competing load on the selected peer during download), and user behavior (the selection criteria a user implements when multiple peers hold a copy of the requested resource, cooperative/non-cooperative behaviors).

In this paper we develop an analytic modeling technique for the analysis of P2P file sharing applications with the aim of providing Quality of Service (QoS) user-perceived measures related with the transfer phase for a given resource; in particular, we provide a method for

---

<sup>1</sup>The bandwidth of channel connecting peers to ISPs.

the computation of transfer time distribution. 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.

Inspired by the work of Anick, Mitra and Sondhi in [3] and the subsequent success of fluid modeling of packet networks (see for instance [6, 18, 19]), we use a fluid approach for modeling the transfer phase in P2P file sharing applications where the transferred resource is represented by a fluid. P2P applications exhibit continuously increasing complexity in several dimensions, such as number of users, number of available resources, etc. Performance analysis of this kind of applications suffers from this complexity and often requires costly numerical methods or model simulations. In this case the fluid approach may contribute to tackle the model solution complexity problem that typically occurs when analyzing discrete state based models.

The model will account for several aspects related to the file transfer in P2P networks; in particular, the impact of resource popularity, s\_peer selection policy, bandwidth characteristics, concurrent downloads and uploads, cooperation level of both c\_peer and s\_peer are included in our model specification.

This paper is an extension of the work presented in [8] where a Fluid Stochastic Petri Net Model of an extremely simplified P2P file sharing application is presented<sup>2</sup>.

The outline of the paper is the following: in Section 2 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 3 briefly summarizes the peculiarities of the stochastic fluid formalism that we use. Section 4 discusses the main issues dominating the transfer time for a resource that are captured by our analytical modeling technique that is illustrated in Section 5 along with a model construction example presented in the Appendix. Some numerical results of the model analysis are presented in Section 6, for instance, we will use the proposed model to investigate the impact of c\_peer bandwidth, overall load on peers participating to the application, and the effects of mechanisms to incentive cooperation among peers. Finally, Section 7 draws conclusions and outlines several development of the current work.

## 2 Related Work

Model based performance evaluation of P2P based application is still in its infancy since there are few analytical models considering P2P networks. In most cases, analysis is carried out through extensive discrete event simulations. For example, [25] proposed a model for the signaling messages in the Gnutella network and concluded that signaling might significantly compromise performance.

---

<sup>2</sup>The work in [8] makes several simplifying assumptions to develop a suitable FSPN model of the system, for instance, uploads are not allowed for c\_peers and downloads are not allowed for s\_peers.

The work in [9] is among the first to model a general P2P system and evaluate its performance. In particular that paper proposed a queuing network based model to explore and illustrate fundamental performances issues of P2P file-sharing systems. This model is applied in three different types 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 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 discrete event simulation approach in [15] for studying complex properties.

In [16] the authors model the request-response process of a file-sharing networks and obtain analytical expression for three performance parameters (delay, jitter and loss probability) as a function of the number of resources indexes in the network and a time-out value for the queries.

In [33] the authors model a file-sharing application, developing a probabilistic model to describe query process and expected query result sizes, they also develop an analytic model to describe system performance.

An analytical model based on age dependent branching processes is developed in [34] where the service capacity of P2P file sharing applications is analyzed by considering a transient regime to characterize the ability of such systems to handle bursty traffic.

The work in [4] is one of the few examples of the use of fluid models to analyze P2P based applications. A fluid model for the performance analysis of the Squirrel cooperative cache system is proposed and studied. To cope with the large number of users that join and leave the cache system randomly, the request streams of the individual nodes are approximated by a fluid flow.

In [7] 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 [7] is based on a use of simulative results combined with some simple analytical considerations.

To properly employ analytical models to investigate properties and performance of the system being represented, model parameters must be set to realistic values obtained from existing measures.

A large body of research on P2P systems so far has emphasized design, traffic measurements, and workload analysis. The proposals presented in [23, 24, 28] studied traces of P2P applications such as Gnutella and Napster. These works focused on characterizing the overall P2P system, e.g., request patterns, traffic volume, traffic categorization and properties of shared online content as well as P2P structure and dynamics, e.g., connectivity and peer behaviors. In particular the papers [23, 24] presents a quantitative evaluations of P2P system behaviors and it is based on a "crawler" to extract topology of Gnutella application level

network and brings to conclude that, although Gnutella is not a pure power-law network, its configuration has benefits and drawbacks of a power-law structure, e.g., resilience to random peer failures.

In [28] 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; furthermore it arises that, despite of the nature of P2P system, there is a clear evidence of client-like or server-like behavior in a significant fraction of the peer populations.

A characterization of P2P traffic and its impact on the underlying network is presented in [29]. In this paper 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 in [11] presents an analysis of a modern P2P multimedia file-sharing workload: a trace-based analysis of Kazaa is presented. The results show that P2P file-sharing workloads are driven by considerably different processes than the classical Web applications. It is also demonstrated that there is significant locality in Kazaa workload, and therefore substantial opportunity for caching to reduce bandwidth consumption. More recent research in the direction of the evaluating P2P systems has focused on performance. Peer selection schemes are evaluated in [21], where measurements are used to optimize the selection of “good peers” and improve the overall system performance.

### 3 Stochastic Fluid Flow Models

A stochastic fluid flow model describes the behavior of a fluid level in a storage device. The input and output rates are assumed to be modulated by a finite homogeneous Markov process. Such models have been used in asynchronous transfer mode (ATM) to evaluate the performance of fast packet switching and in manufacturing systems for the performance of producers and consumers coupled by a buffer. There is a large number of papers dealing with the analysis of stochastic fluid flow models. Most of these papers consider such models in stationary regime. Anick et al. [3] and Kosten [17] analyzed the fluid model for several on-off input sources controlled by a two-state homogeneous Markov process. Mitra [20] generalized this model by considering multiple on-off inputs and outputs.

We describe a stochastic fluid model with an infinite (fluid) buffer for which the input and output rates are controlled by a homogeneous Markov process  $M = \{M_s, s \geq 0\}$  on the finite state space  $\mathcal{S}$  with infinitesimal generator  $\mathbf{Q}$  and initial probability distribution  $\boldsymbol{\pi}_0$ . The number of states is denoted by  $|\mathcal{S}|$ . The amount of fluid in the buffer at time  $\tau$  is denoted by  $x_\tau$ . The pair  $(M_\tau, x_\tau)$  forms a Markov process having a pair of discrete and continuous

states. Let  $r_i^+$  be the input rate and  $r_i^-$  the output rate when the Markov process  $M$  is in state  $i$ . We denote by  $r_i$  the effective input rate of state  $i$ , that is  $r_i = r_i^+ - r_i^-$ .

For each state  $i$  of the Markov process we compute  $\pi_i(\tau, x)$  which is the probability density of finding the process in discrete state  $i$  with fluid level  $x$  at time  $\tau$ . If we call  $q_{ij}(x)$ ,  $i \neq j$ , the transition rate from state  $i$  to state  $j$  when the fluid level is equal to  $x$ , and define  $q_{ii}(x) = -\sum_{j \neq i} q_{ij}(x)$ , we have the following partial differential equations (see [31] for details):

$$\frac{\partial \pi_i(\tau, x)}{\partial \tau} + \frac{\partial (r(i, x)\pi_i(\tau, x))}{\partial x} = \sum_{j \in S} (\pi_j(\tau, x)q_{ji}(x)), \quad (1)$$

In order to solve Equation (1), boundary and initial condition must be specified. Boundary conditions can become quite complex when the fluid rate can become negative in some state  $i$  ( $r_i \leq 0$ ). In this case some probability mass is formed at the boundary and it must be explicitly considered as in [12]. In the special case in which  $r_i > 0, \forall i$ , then the boundary conditions become quite simple:

$$\pi_i(\tau, 0) = 0. \quad (2)$$

If we consider that the initial fluid level is 0 in every state, and we specify  $\pi_0^{(i)}$  the probability that the initial discrete marking is state  $i$ , then the initial condition can be specified as:

$$\pi_i(0, x) = \delta(x)\pi_0^{(i)}, \quad (3)$$

where  $\delta(x)$  is the Dirac's delta function.

The first terms of the left hand side of Equation (1) accounts for the time elapsed in the state (i.e., in discrete state  $i$  with fluid level  $x$ ). The term  $r(i, x)$  represents the actual flow rate in discrete state  $i$  (for each discrete state  $i$ ). From this it follows that the second term of Equation (1) accounts for the infinitesimal changes of the fluid level, whose rate is expressed by the term  $r(i, x)$ . The term on the right hand side, accounts for both the transitions into and out from the state.

Several solution algorithm have been proposed in the literature to solve those equations. In [12] *positive semidiscretization*, together with *uniformization* has been proposed as a general solution algorithm. An efficient transient analysis technique for fluid models with constant fluid-rates is proposed in [30]. Transient analysis of second order stochastic fluid model has been considered in [32]. Steady state analysis of first order bounded models has been considered in [10].

## 4 Parameters that Dominate the Transfer Phase in P2P Networks

In general, in P2P file sharing applications, a successful search provides a list of peers holding a copy of the requested resource. In the following we denote the peer that requests the

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 1: Distribution of the user bandwidths (from [28] voluntarily reported by Napster users)

resource as the *client-like peer* ( $c\_peer$ ) and the peers holding a copy of the requested resource as the *server-like peers* ( $s\_peer$ ). For each  $s\_peer$  and copy of the requested resource the P2P applications also provide additional information such as the bandwidth between the  $s\_peer$  and its ISP, the number of  $c\_peers$  that are using this  $s\_peer$  the quality of the resource, and other information that help the  $c\_peer$  to choose the  $s\_peer$  to download the resource.

In the following we discuss the parameters, the characteristics of the P2P file sharing applications, and the user behaviors that influence the transfer phase duration.

- *Network Characteristics*: the rate at which files can be downloaded from a  $s\_peer$  depends on the bottleneck bandwidth between the  $c\_peer$  and the chosen  $s\_peer$ , the available bandwidth, and the latency along the path connecting the two peers. In [28] measure 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 range over several orders of magnitude across the peers of the system. Table 1 reports these measures for the distribution of connection bandwidths. It is important to point out that the measures reported in Table 1, 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 thus the results presented in this paper are also computed for a different bandwidth distribution where the fraction of high-speed connections is increased while at the same time the fraction of slower dial-up connections is decreased.
- *Application Characteristics*: after the  $c\_peer$  chooses a  $s\_peer$  (or more  $s\_peers$  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  $s\_peer$  and also depend on the maximum number of concurrent uploads/downloads.

In general, P2P applications implement sharing bandwidth policies among the different `c_peers` that download resource(s): in some case the `s_peer` equally shares the available download bandwidth among the `c_peers`, in other cases the sharing policy depends on some parameters that account for the participation level of the `c_peer`. Furthermore, bandwidth fluctuations are also influenced by the maximum number of downloads operations allowed by the `s_peer`. The speed at which the `c_peer` downloads the requested file also depends on the possibility that the file can be downloaded in pieces or “chunks” from several different `s_peer` (for instance, Kazaa allows this possibility).

- *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, 27]) show 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, typically MP3 files, that are less than 10 MB, *medium-size* resources, 10 to 100 MB that correspond to small-medium video files, and *large-size* resources over 100 MB, that correspond to large video files.

The quality of the resource is a factor that has a direct translation into the size characteristics of resource and can be used as a parameter to select a particular `s_peer` for downloading. In the case of a MP3 file the quality of the resource is intended in terms of compression bitrate, in the case of a still image coded using the JPEG standard the quality is intended in terms of the compression parameter. Quality information for other resource types can be easily defined and it is retrieved from the search results and provided as an attribute of the copy of the requested resource.

The influence of the resource popularity, i.e., the number of peers that hold a copy of the resource, on the transfer rate is quite clear. If the `c_peer` is looking for a very popular resource, than the probability that a copy of this resource is held by a `s_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 `s_peer` holding a very popular resource is overloaded (because there are many `c_peers` that require its resource(s)) increases with the resource popularity.

- *User Behavior*: When multiple `s_peers` hold a copy the requested resource, the user has to select the actual `s_peer` from which downloading the resource. The choice could be based on the `s_peer` with the fastest connection, on the `s_peer` with the lowest load, on the quality of the resource, it could be a random selection, etc.

Another important issue related to the user behavior concerns the cooperation among users. Without cooperation users of such applications suffer long download delays, if they are able to download at all [2]. Usually, existing P2P applications rely on altruism, which inevitably fails when participants show some selfish behavior. In order to prevent these behavior, many P2P protocols applies a user-managed incentive system. In Section 6 we investigate the effects on the user-perceived performance measures of some simple incentive mechanisms.



## 5 Model description

The goal of the proposed model is to compute the transfer time distribution for the download of a file of a given length  $s$ , available on  $n$  peers, from a  $c\_peer$  with a given bandwidth  $cb$ . We will denote this distribution as  $F_t(t|cb, s, n)$ ; it represents the probability that a  $c\_peer$  with bandwidth  $cb$  can download a resource of length  $s$  that is available on  $n$  peers in less than  $t$  units of time.

In the following we always consider the case of a particular  $c\_peer$  that transfers a resource  $r_s$  from a given  $s\_peer$ . We will address these entities respectively as *tagged c\_peer*, *tagged transfer* and *tagged s\_peer*. We analyze the tagged transfer by making transient analysis of the model: we assume that the tagged  $c\_peer$  starts downloading the resource  $r_s$  at time 0, and stays in the system until it has finished downloading it. The tagged  $c\_peer$  leaves the system as soon as it has finished the transfer and we do not consider queuing of requests before starting the resource transfer. For these assumptions, we model only the interference caused by other peers.

Consider the file transfer of given resource  $r_s$  from a  $s\_peer$  of bandwidth  $sb$ . Two entities interact in the file transfer: a peer which acts as a  $c\_peer$ , and another peer that acts as a  $s\_peer$ . The  $c\_peer$  downloads resource  $r_s$ , and the  $s\_peer$  uploads resource  $r_s$ . Since we are in a P2P network, both the  $s\_peer$  and the  $c\_peer$  can handle more simultaneous uploads and downloads. The download time distribution depends on the bandwidth of both peers, on the number and on the type of concurrent uploads and downloads at both peers, and on the network load.

The  $c\_peer$  behavior, the  $s\_peer$  behavior and the network load, can be described using continuous time, discrete state stochastic processes. We call *c\_peer process* the stochastic process that describes the  $c\_peer$  behavior, and  $\mathcal{S}_c$  its state space. In the same way we define the *s\_peer process* (state space  $\mathcal{S}_s$ ) and *the network process* (state space  $\mathcal{S}_n$ ). Each state  $s_c \in \mathcal{S}_c$  and  $s_s \in \mathcal{S}_s$  takes into account the number of concurrent uploads, downloads and their actual speed. Each  $s_n \in \mathcal{S}_n$  considers a different load of the network.

If we denote by  $f(s_c, s_s, s_n)$  the transfer rate at which resource  $r_s$  is downloaded when the  $c\_peer$ , the  $s\_peer$  and the network are in state  $s_c, s_s, s_n$  respectively, then the number of bytes  $x(\tau)$  downloaded in time  $\tau$  are:

$$x(\tau) = \int_0^\tau f(s_c(t), s_s(t), s_n(t)) dt.$$

During the download the entities change states, and this is denoted by writing  $s_c(t), s_s(t), s_n(t)$ .

In the following in Section 5.1 we first present the fluid model and the solution technique to compute the transfer time distribution given that both the  $s\_peer$  bandwidth and the initial states are known. A detailed model construction example is provided in the Appendix. Section 5.2 describes the technique to remove the dependency on the initial state, and Section 5.3 describes the technique to remove the dependency on the  $s\_peer$  bandwidth. Finally, Section 5.4 proposes a general framework based on the concept of peer classes to extend the modeling approach to deal with several interesting scenarios.

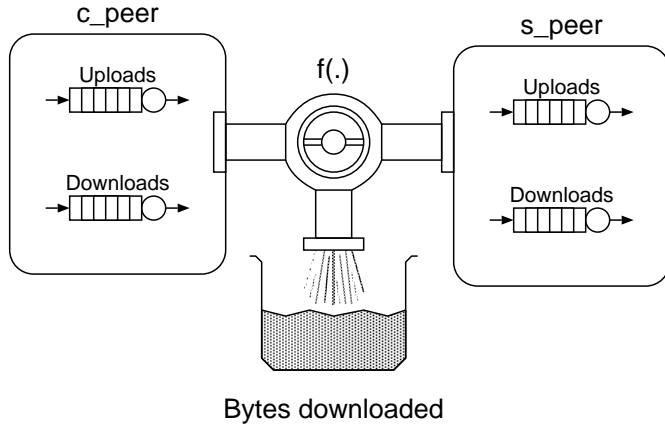


Figure 1: The considered model

## 5.1 The Stochastic Fluid Model

In our analysis we assume that communication bottlenecks can only happen at the `c_peer` or at the `s_peer` (i.e. only the first and last mile), and that the backbone is never a bottleneck. In other words, we assume that  $f(s_c, s_s, s_n) = f(s_c, s_s, s'_n), \forall s_n, s'_n \in \mathcal{S}_n$ . Thus we drop the dependency on the network state  $s_n$ , and we specify the transfer rate simply as:  $f(s_c, s_s)$ . We assume that that actual transfer rate is just the minimum between two independent terms: the maximum upload rate at the `c_peer`  $f_c(s_c)$ , and the maximum download rate at the `s_peer`  $f_s(s_s)$ , that is:

$$f(s_c, s_s) = \min(f_c(s_c), f_s(s_s)). \quad (4)$$

We also assume that the tagged `c_peer` always selects a peer that is immediately available for downloading. We chose to represent both the `c_peer` process and the `s_peer` process with two independent components each: the *download component*, and the *upload component*. Both components are modeled using finite capacity queues, as depicted in Figure 1. The queue representing the download component at the `c_peer`, and the queue representing the upload component at the `s_peer`, have at least one customer (the tagged transfer of the requested resource  $r_s$ ). The other customers represent respectively:

- The other resources that our tagged `c_peer` is downloading in parallel with resource  $r_s$  (*c\_peer download component*).
- The resources that our tagged `c_peer` is uploading when acting as a `s_peer` for other peers. (*c\_peer upload component*).
- The other resources that our tagged `s_peer` is uploading in parallel with resource  $r_s$  (*s\_peer upload component*).
- The resources that the tagged `s_peer` is currently downloading from other peers (*s\_peer download component*).

In the following we will refer to this queues by using the subscripts  $cd$ ,  $cu$ ,  $su$ , and  $sd$  respectively.

If we denote by  $\mathcal{S}_{cu}$  and  $\mathcal{S}_{cd}$  the states spaces of the upload and download components at the c\_peer, and  $\mathcal{S}_{su}$  and  $\mathcal{S}_{sd}$  the states spaces of the upload and download components at the s\_peer, we have that  $\mathcal{S}_c = \mathcal{S}_{cu} \times \mathcal{S}_{cd}$  and  $\mathcal{S}_s = \mathcal{S}_{su} \times \mathcal{S}_{sd}$ .

In this paper we choose to model all the queuing components using  $M/H_2/\infty/K$  queues, where  $H_2$  refers to a two stages hyper-exponential distribution.

The inter-arrival distribution is assumed to be exponential. We also consider that the arrival rate may depend on the s\_peer bandwidth, and may be different for the upload or the download queue, as in [28]. In this way the queues representing the four model components have each one a different arrival rate. We denote these rates with  $\lambda_{cu}(cb)$ ,  $\lambda_{cd}(cb)$ ,  $\lambda_{su}(sb)$ ,  $\lambda_{sd}(sb)$ . Each of these rates is derived as a fraction of a reference value  $\lambda_0$  obtained from the measures reported in [23].

The choice of the two stage hyper-exponential distribution for the service time distribution of the four queues stems from available measures [11] that provide statistics on the distribution of session times for downloads. In particular, this study highlights that session times exhibit high variability (because of the large heterogeneity in resource sizes); the two-stage hyper-exponential distribution is a very simple Markovian representation of distributions whose coefficient of variation is greater than one and it allows to keep the discrete state space of the stochastic fluid model as small as possible. It would be by no means difficult to use more complex phase-type Markovian approximations of heavy-tailed distributions to represent variability in session times; the impact of this choice on the model we developed would be an increase of number of the discrete states.

$K$  represents the maximum number of simultaneous interfering transfers (upload or download) allowed by each component. This maximum is assumed to be dependent on the bandwidth, to be different for the upload and the download queue, and to exclude the tagged transfer. We denote these different values with  $M_{cu}(cb)$ ,  $M_{cd}(cb)$ ,  $M_{su}(sb)$ ,  $M_{sd}(sb)$ . Note that in the case of  $M_{cd}(cb)$ ,  $M_{su}(sb)$ , the actual maximum is one unit bigger for the assumption that the tagged transfer is excluded from the model.

We assume that the available bandwidth at the s\_peer for download is equally shared among all the competing uploads and downloads if there is no distinction between upload and download bandwidths (e.g. Cable Modem, Ethernet). On the other hand, if uploading bandwidth and downloading bandwidth are separated (e.g., dial-up and asymmetric DSL connections) the available bandwidth at the s\_peer for download is equally shared among all competing downloads. The same rule applies at the c\_peer for uploads.

According to the previous considerations we can write:

$$f_s(s_{su}, s_{sd}) = \begin{cases} \frac{sb}{\#s_{su} + \#s_{sd} + 1} & \text{if downloading and uploading} \\ & \text{bandwidths are not separated} \\ \frac{sb_{\text{upld}}}{\#s_{sd} + 1} & \text{if downloading and uploading} \\ & \text{bandwidths are separated} \end{cases} \quad (5)$$

where  $\#s_{su}$ , and  $\#s_{sd}$ , represent the number of customers in queues  $su$  and  $sd$ ;  $sb$  is the s\_peer bandwidth; and  $sb_{\text{upld}}$  is the s\_peer uploading bandwidth (when the uploading and downloading bandwidths are separated). We can write the same kind of equation for the c\_peer

$$f_c(s_{cu}, s_{cd}) = \begin{cases} \frac{cb}{\#s_{cu} + \#s_{cd} + 1} & \text{if downloading and uploading} \\ & \text{bandwidths are not separated} \\ \frac{cb_{\text{dwnl}}}{\#s_{cd} + 1} & \text{if downloading and uploading} \\ & \text{bandwidths are separated} \end{cases} \quad (6)$$

where  $\#s_{cu}$ , and  $\#s_{cd}$ , represent the number of customers in the queue  $cu$ ,  $cd$ ;  $cb$  is the c\_peer bandwidth; and  $cb_{\text{dwnl}}$  is the c\_peer downloading bandwidth (when the uploading and downloading bandwidths are separated). In Equations (5) and (6) the tagged transfer is represented by adding one to the denominator. By Equation (4), we can define:

$$f((s_{cu}, s_{cd}), (s_{su}, s_{sd})) = \min(f_c(s_{cu}, s_{cd}), f_s(s_{su}, s_{sd})).$$

where  $f_c(s_{cu}, s_{cd})$  and  $f_s(s_{su}, s_{sd})$  are defined by Equations (5) and (6).

With the previous assumptions, the underlying stochastic process of the described model is a stochastic fluid model. The state space is defined as:  $\mathcal{S}_c \times \mathcal{S}_s = \mathcal{S}_{cu} \times \mathcal{S}_{cd} \times \mathcal{S}_{su} \times \mathcal{S}_{sd}$ . If we call  $\mathbf{Q}_{cu}$ ,  $\mathbf{Q}_{cd}$ ,  $\mathbf{Q}_{su}$  and  $\mathbf{Q}_{sd}$  respectively the generator matrices of the queuing networks corresponding to the c\_peer upload component, c\_peer download component, s\_peer upload component and s\_peer download component, then the generator matrix of the stochastic fluid model can be defined as:

$$\mathbf{Q} = \mathbf{Q}_{cu} \oplus \mathbf{Q}_{cd} \oplus \mathbf{Q}_{su} \oplus \mathbf{Q}_{sd},$$

that is, the Kronecker sum of the generator matrix of the single components. The rate matrix of the stochastic fluid model, can be defined as a diagonal matrix  $\mathbf{R} = \text{diag}(\hat{f}(i))$ , where:

$$\hat{f}(i) = f((s_{cu}[i], s_{cd}[i]), (s_{su}[i], s_{sd}[i])),$$

and  $s_l[i]$  represents the part related to the  $l$  component of state  $i$ , with  $l \in \{cu, cd, su, sd\}$ . Figure 2 shows a graphical representation of the stochastic fluid model. If we denote as  $\pi_j(\tau, x)$  the probability density of having  $x$  unit of fluid at time  $\tau$  in discrete state  $j$ , i.e., the probability that  $x$  bytes of a resource have been downloaded at time  $\tau$  in state  $j$ , we can rewrite Equation (1) in matrix / vector notation:

$$\frac{\partial \boldsymbol{\pi}(\tau, x)}{\partial \tau} + \frac{\partial \boldsymbol{\pi}(\tau, x)}{\partial x} \mathbf{R} = \boldsymbol{\pi} \mathbf{Q}, \quad (7)$$

where  $\boldsymbol{\pi}(\tau, x)$  is a vector whose components correspond to  $\pi_j(\tau, x)$ .

The resource transfer time is conditioned by the number of competing peers on both the s\_peer and on the c\_peer when the transfer starts. This problem will be dealt with in Section 5.2. In this section, we assume that the initial states of the c\_peer and of the s\_peer are specified

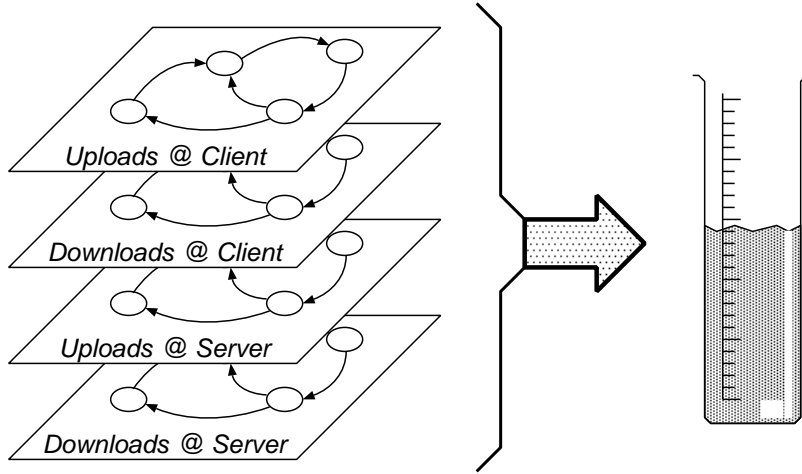


Figure 2: The Markov Process underlying the model

as a model parameter, denoted as  $\boldsymbol{\pi}_0$ . The parameter  $\boldsymbol{\pi}_0$  is a probability distribution on the model state space  $\mathcal{S}_{cu} \times \mathcal{S}_{cd} \times \mathcal{S}_{su} \times \mathcal{S}_{sd}$  (the  $j$ -th component of vector  $\boldsymbol{\pi}_0$  that corresponds to the probability of state  $(s_{cu}[j], s_{cd}[j]), (s_{su}[j], s_{sd}[j])$  is denoted as  $\pi_0(j)$ ). Since  $\hat{f}(j) > 0$  for any state  $j$ , and since the fluid place is unbounded the initial and boundary conditions are:

$$\begin{aligned}\boldsymbol{\pi}(0, x) &= \delta(x)\boldsymbol{\pi}_0, \\ \boldsymbol{\pi}(\tau, 0) &= \mathbf{0}.\end{aligned}\tag{8}$$

Many of the terms in Equation 7 depend on the c\_peer bandwidth  $cb$  and on the s\_peer bandwidth  $sb$ . In particular, matrices  $\mathbf{Q}$  and  $\mathbf{R}$  depends on both the  $sb$  and  $cb$  since the arrival rate of the queues and the definition of rate function depend both on  $sb$  and  $cb$ . By explicitly showing these dependencies, we can denote the solution of Equation (7) as  $\boldsymbol{\pi}(\tau, x, sb, cb, \boldsymbol{\pi}_0)$  for a given combination of parameters  $sb$ ,  $cb$  and  $\boldsymbol{\pi}_0$ .

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  $\tau = t$ , and can be computed as:

$$F_t(t|sb, cb, s, \boldsymbol{\pi}_0) = \int_s^\infty \bar{\pi}(\tau, x, sb, cb, \boldsymbol{\pi}_0) dx \Big|_{\tau=t}\tag{9}$$

where  $\bar{\pi}(\tau, x, sb, cb, \boldsymbol{\pi}_0) = \boldsymbol{\pi}(\tau, x, sb, cb, \boldsymbol{\pi}_0)\mathbf{1}$ , and  $\mathbf{1}$  is a unit vector with a number of components equal to the number of discrete states of the model.  $\bar{\pi}(\tau, x, sb, cb, \boldsymbol{\pi}_0)$  represents the probability density of the fluid level regardless of the discrete state.

For the sake of simplicity we chose to represent both the c\_peer process and the s\_peer process with two independent components each: the download and the upload component. Dependencies among the four queueing processes may be captured by suitably modifying the modulating discrete part of the stochastic fluid model. In this case, the modifications may

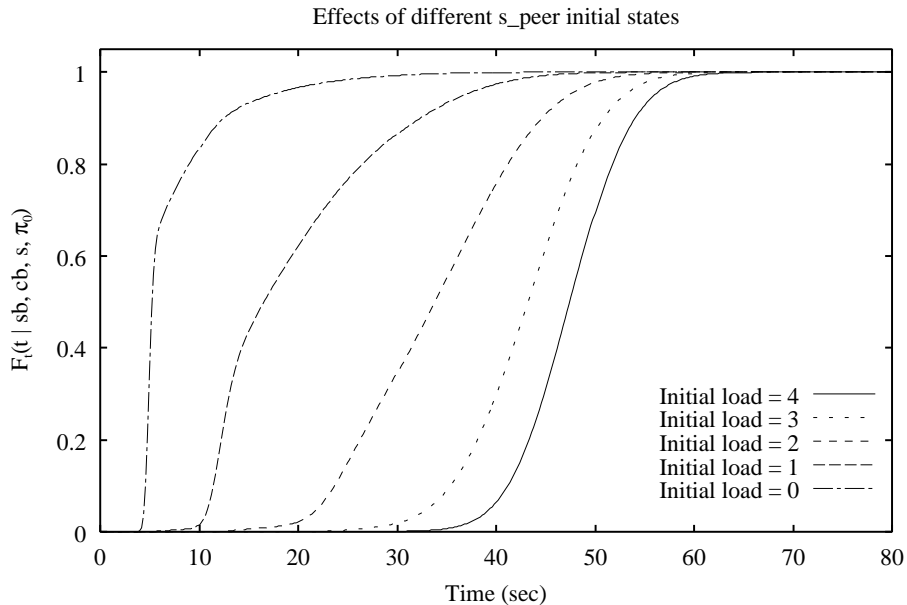


Figure 3: The dependency of the transfer time from the initial load

prevent the possibility of describing the matrix  $\mathbf{Q}$  by means of Kronecker sum of the generator matrix of the single components.

## 5.2 Removing the initial state dependency

The initial state of each peer involved in a file transfer can have a significant impact on the time necessary to complete it. This is more evident in the case of shorter files. For example, Figure 3 shows the transfer time distribution for different initial loads, calculated for a 112 KB JPEG image from a DSL peer using a DSL connection, when there are no simultaneous transfers on the c\_peer and the s\_peer allows five simultaneous uploads but does not perform any downloads. The download time of the resource is shorter when the system is less loaded at the beginning of the transfer. The same consideration are enhanced if we consider the full P2P scenario where both the c\_peer and the s\_peer can have many simultaneous uploads and downloads.

From these consideration it is clear that it is important to determine correctly the initial state of the model. The model, as presented in Figure 1, is composed by four queuing networks. We assume that when the download starts, the tagged c\_peer finds all the four queues in their steady state. In this manner, the initial state of the model is computed by determining the steady state solution of the four finite capacity queues. In particular we determine:

$$\begin{cases} \pi_l^0 \mathbf{Q}_l = 0 \\ \pi_l^0 \mathbf{1}_l = 1 \end{cases} \quad (10)$$

with  $l \in \{cu, cd, su, sd\}$ , and where  $\mathbf{1}_l$  is a vector of size  $|\mathcal{S}_l|$  with all the components equal to one. Then we define the initial state of the whole model  $\pi_0$  as:

$$\pi_0 = \pi_{cu}^0 \otimes \pi_{cd}^0 \otimes \pi_{su}^0 \otimes \pi_{sd}^0,$$

where the  $\otimes$  operator indicates that each component of the left vector is multiplied by all the components of the right vector. Note that since  $\mathbf{Q}$  depends on both the c\_peer bandwidth  $cb$  and the s\_peer bandwidth  $sb$ , also  $\pi_0$  depends on  $cb$  and  $sb$ . We do not explicitly show this dependency by writing  $\pi_0(cb, sb)$  for sake of simplicity. We denote the file transfer distribution, after the dependency on the initial state has been removed, by:

$$F_t(t|sb, cb, s) = F_t(t|sb, cb, s, \pi_0(cb, sb))$$

### 5.3 Removing the dependence on the s\_peer bandwidth

In P2P environment, after a successful search of the requested resource, the c\_peer obtains a list of the peers holding a copy of the file. This list provide a set of information regarding both the peers (i.e. their bandwidth, queue status, etc.) and the resource (i.e. bitrate, size, etc.). The c\_peer can use these information to select the peer, or a set of peers in case of multiple download, from which get the resource. The choice could be based on the peer with the fastest connection, or on the peer with the lowest load, or on the resource with the highest bitrate, it could be a random selection, etc.

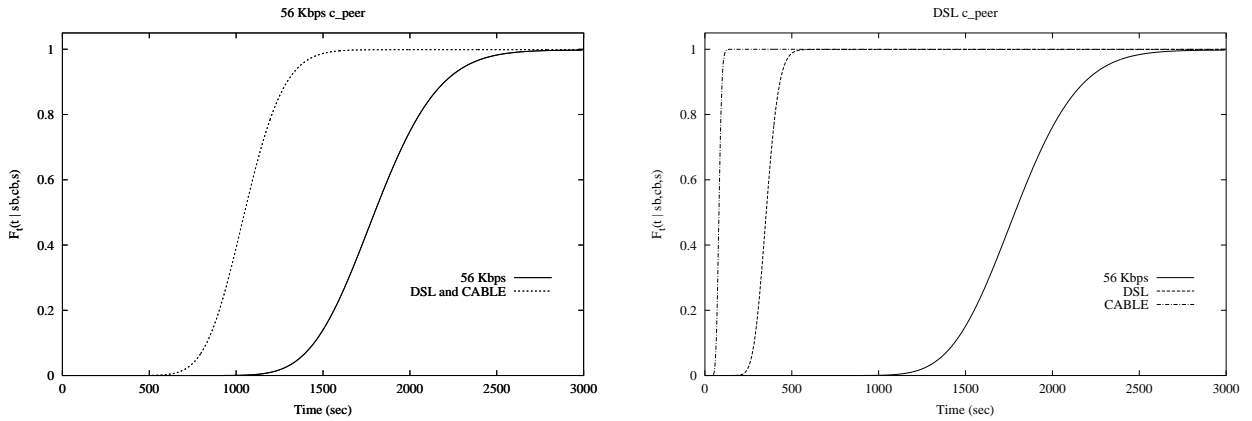


Figure 4: The effect of s\_peer bandwidth on the distribution of the transfer time (4 MB resource size) for a modem 56 Kbps c\_peer (left graph) and for a DSL c\_peer (right graph).

What is important to take into account is that the selection policy can impact on the file transfer performance. For instance the selection of the peer bandwidth affects the transfer time distribution. The impact is much stronger if the `c_peer` peer has a fast connection, such as DSL, Cable, T1 and T3. We performed some experiment to verify the dependency of the performance from the `s_peer` bandwidth. We considered two scenarios with two different `c_peer` bandwidth, modem 56 Kbps and DSL, and for each case we calculated the file transfer time distribution for different `s_peer` bandwidths. In particular, we made these experiments with the values 56 Kbps, DSL and Cable. The results reported in Figure 4, relative to the transfer of a resource with size  $s$  equal to 4 MB, show that when the `c_peer` has a 56 Kbps bandwidth the transfer time is conditioned by both its and the `s_peer` bottleneck. In this case, when the `s_peer` bandwidth is faster (DSL or Cable) the transfer performance are limited by the `c_peer` bandwidth; when the `s_peer` bandwidth is 56 Kbps the transfer rate depends on the simultaneous operations that the two peers are performing during the tagged transfer. When the `c_peer` bandwidth is DSL instead, the performance depends always on the `s_peer` bandwidth; in the case of DSL and Cable bandwidth the role of bottleneck depends on the probability of having simultaneous transfer on both the peers. In the case of 56 Kbps the `s_peer` behaves always like bottleneck and the transfer time depends on its bandwidth. In our model we assumed that the selection peer policy was based on the largest bandwidth. To model this selection policy we made the same consideration reported in [8]. We consider that the bandwidth of the peers holding a resource are i.i.d. random variables. For the distribution of the bandwidths we consider the one reported from Napster's users and described in Table 1.

According to this distribution, we define a combinatorial manipulation that models the selection of the peer with the highest bandwidth. To this end, we introduce one of the model parameters that is the resource popularity: it is defined as the number of available resources for the tagged transfer, i.e., the number of peers that hold a copy of the content to be transferred. We denote the popularity of the resource identified by the tagged transfer as  $N$ .

Our model assumes that when the resource is available on only one `s_peer`, the probability  $P(SB = sb)$  is exactly the one reported in Table 1. When the number of available resources is greater than one, the probability to select a peer with a 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 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 1.

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:

$$\begin{aligned}
 P(SB = sb|N = n) &= P(SB \leq sb|N = n - 1)P(SB = sb) + \\
 &P(SB = sb|N = n - 1)P(SB < sb),
 \end{aligned}
 \tag{11}$$

where



$$\begin{aligned}
P(SB \leq sb | N = n - 1) &= \begin{cases} \sum_{\hat{b} \leq sb} P(SB = \hat{b} | N = n - 1) & \text{if } n > 1, \\ 1 & \text{if } n = 1, \end{cases} \\
P(SB = sb | N = n - 1) &= 0 \quad \text{if } n = 1, \\
&\text{and} \\
P(SB < sb) &= \sum_{\hat{b} < sb} P(SB = \hat{b}).
\end{aligned}$$

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

Figure 5 depicts the probability  $P(SB = sb | N = n)$  as function of  $sb$  and  $n$ . 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 s\_peer with fastest bandwidth increases with the number of s\_peers holding a copy of the resource.

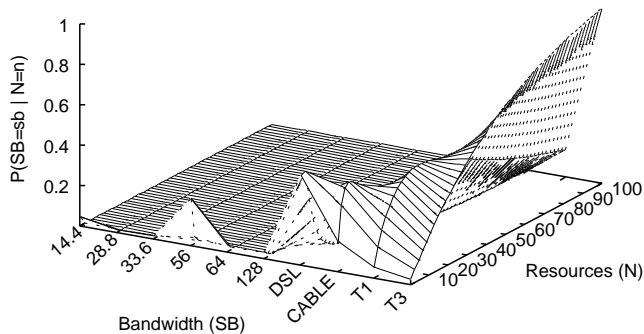


Figure 5:  $P(SB = sb | N = n)$  distribution as function of  $SB$ , and  $N$ .

Using Equation 11, the initial state computed in Section 5.2, and Equation 9, we can compute:

$$F_t(t|cb, s, n) = \sum_{sb} F_t(t|sb, cb, s) P(SB = sb | N = n). \quad (12)$$

## 5.4 Model Extensions

The proposed model can be easily extended to consider more complex scenarios. In this section we sketch two possible extensions: one considers the case where a resource is downloaded in parallel from more than one s\_peer, and the other considers general peer characteristics that might allow, for instance, the modeling of heavily loaded peers known as "hot spots".

### 5.4.1 Parallel download

In order to consider parallel downloads, the model must be extended to consider more than one s\_peer. For example, Figure 6 shows how the model can be extended to consider two s\_peers ( $s_1$  and  $s_2$ ). Each s\_peer can have its own characteristics (bandwidth, capacity and load). The state space of the model will now be composed of 3 components:  $\mathcal{S}_c$  which represents the c\_peer, and  $\mathcal{S}_1$  and  $\mathcal{S}_2$  which represents the two s\_peers. In this case, the actual download speed should depend on the state of both s\_peers, as proposed in Equation 13

$$f(s_c, s_{s_1}, s_{s_2}) = \min(f_c(s_c), f_{s_1}(s_{s_1}) + f_{s_2}(s_{s_2})). \quad (13)$$

Also in this case, each component could be described using two queueing network. The total state space will then be expressed as the cartesian product of the 6 sub-components ( $\mathcal{S} = \mathcal{S}_{cu} \times \mathcal{S}_{cd} \times \mathcal{S}_{su_1} \times \mathcal{S}_{sd_1} \times \mathcal{S}_{su_2} \times \mathcal{S}_{sd_2}$ ), and the infinitesimal generator of the fluid model as the Kronecker sum of six matrices ( $\mathbf{Q} = \mathbf{Q}_{cu} \oplus \mathbf{Q}_{cd} \oplus \mathbf{Q}_{su_1} \oplus \mathbf{Q}_{sd_1} \oplus \mathbf{Q}_{su_2} \oplus \mathbf{Q}_{sd_2}$ ).

In addition, when considering parallel download from more than one s\_peer, one fluid model must be produced for every pair of values for the s\_peer bandwidths  $sb_1$  and  $sb_2$ . The solution of the fluid model provides in this case  $F_t(t|sb_1, sb_2, cb, s)$ , that is, the download time distribution of a file of length  $s$ , for a c\_peer whose bandwidth is  $cb$  that downloads the resource in parallel from two s\_peers of bandwidth  $sb_1$  and  $sb_2$ . The final distribution can be then computed as:

$$F_t(t|cb, s, n) = \sum_{sb_1} \sum_{sb_2} F_t(t|sb_1, sb_2, cb, s) P(SB_1 = sb_1, SB_2 = sb_2 | N = n). \quad (14)$$

where  $P(SB_1 = sb_1, SB_2 = sb_2 | N = n)$  is now the probability of finding a resource with popularity  $n$  on two s\_peers whose bandwidths are  $sb_1$  and  $sb_2$ , respectively. The parallel download extension requires more computational efforts since both the number of discrete states of the fluid model and the number of sub-models which must be considered grow exponentially with the number of s\_peers.

### 5.4.2 More general peer characteristics

In the proposed model, the peers are actually partitioned in classes characterized by their bandwidth value, i.e., every parameter of the fluid model depends only on the peer bandwidth. To consider a more general scenario where the peer characteristics depend also on other parameters, we could consider the peers as partitioned into a number  $u$  of general peer classes  $C = \{c_1, c_2, \dots, c_u\}$ . For instance, a peer class could be used to represent peers with high bandwidth, a large number of resources, and a heavy request load (hot spots), or to represent peers with dial-up connections, no resources to share, and no request load (free raiders).

For each class  $c$ , we define the arrival rate  $\lambda(c)$ , the maximum upload and download capacity  $K_u(c)$  and  $K_d(c)$ , the maximum bandwidth for download and upload  $sb_u(c)$  and  $sb_d(c)$ , and the actual bandwidth function  $f_{s(c)}(s_s(c))$ . Using the proposed extension, we

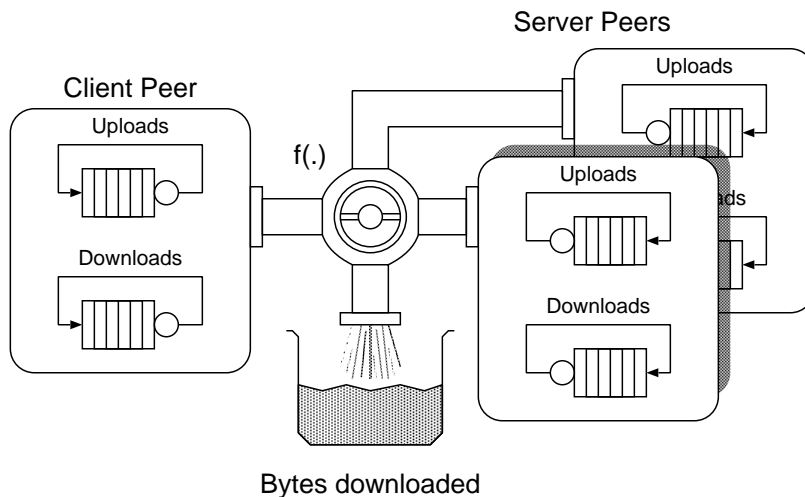


Figure 6: Extension to consider two s\_peers

are able to determine the download time distribution of a file of length  $s$  from a c\_peer of bandwidth  $cb$  from a s\_peer of class  $c$ , which is referred to as  $F_t(t|c, cb, s)$ .

Furthermore, in the original model, the queuing time distribution at the s\_peer has been ignored, assuming that the c\_peer always found the s\_peer available. The model could be extended to remove this assumption, if we are able to compute (using for example a simple queuing model) the probability density of waiting time distribution at a s\_peer of class  $c$  that we denote as  $w(t|c)$ . Generalizing the approach used to derive Equation 11, we can assume we are able to determine the distribution of the probability of finding the tagged resource on a s\_peer of class  $c$ , given the resource popularity  $n$ , i.e.,  $P(C = c|N = n)$ ; in this case, we can compute the final transfer time distribution as:

$$F_t(t|cb, s, n) = \sum_c (F_t(t|c, cb, s) * w(t|c)) P(C = c|N = n). \quad (15)$$

where  $*$  represents the convolution operator.

## 6 Numerical analysis

In this section we present numerical results obtained from the transient analysis of the stochastic fluid model we developed in the previous section. The goal of this presentation is to outline the potentialities of the approach we adopted as well as to investigate interesting issues related to the effect of incentive mechanisms on the user cooperation. In particular, we use the model to compute the transfer time distribution of a resource as function of the c\_peer bandwidth, the overall load on peers taking part in the P2P application, and the interaction between the user cooperation level and the incentive mechanism of the P2P application protocol.

Bandwidths	%
56 Kbps	6%
DSL	23%
Cable	64%
T1	5%
T3	2%

Table 2: User bandwidth distribution for the scenario n. 2

In the following we present numerical experiments derived for two different scenarios. These scenarios are characterized by two different user bandwidth distributions. For the scenario n. 1 we assume that the user bandwidth distribution is the one presented in Table 1. In this case, the user bandwidth distribution is derived from measure-based results presented in [28]. These results have been presented in 2002 and then they can be considered a reasonable “picture” of the last-mile connections at that time.

For the scenario n. 2 we assume that the user bandwidth distribution is the one presented in Table 2. This scenario aims to investigate the increasing of number of peers with high-speed bandwidth connections that replace the slower dial-up connections.

Our model has several parameters whose values have to be assigned to represent realistic environments. For some of them we have been able to exploit published data to assign meaningful and realistic values. Unless otherwise stated, all results have been obtained by using a value for  $\lambda_0$  equal to 7.0 (this value is derived from the measures reported in [23]). The values of  $\lambda_{cu}(cb)$ ,  $\lambda_{cd}(cb)$ ,  $\lambda_{su}(sb)$ , and  $\lambda_{sd}(sb)$  have been defined as a fraction of  $\lambda_0$  as:  $\lambda_{cu}(b) = \lambda_{su}(b) = \lambda_0 \cdot w_u(b)$  and  $\lambda_{cd}(b) = \lambda_{sd}(b) = \lambda_0 \cdot w_d(b)$ . Table 3 reports the actual values we used and that we derived from [28]. The values of the maximum number of allowed uploads have been set as a function of the peer bandwidth as reported in Table 3; for this model parameter we could not find reported measures and decided to assign values to follow the intuition that the higher the bandwidth the higher the number of allowed uploads. Table 3 also reports the values of the two-stages hyper-exponential service distribution that we obtained to fit the available measures [11] that provide statistics on the distribution of session times.

From the transfer time distribution we decided to extract three performance parameters: the average, the 50<sup>th</sup> and the 90<sup>th</sup> quantile of the distribution. We chose these indexes as they provide the user with an estimation of the probability to complete the resource transfer within user defined time thresholds.

Extensive validation of the model results for our modeling technique shares the same difficulty of previous work on analytical models for P2P networks [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 and detailed discrete-event simulations of realistic P2P file sharing applications would have a prohibitive programming and computational cost. Nevertheless, we performed (not shown) simple validations by comparing the

b	$M_{cu}(b), M_{su}(b)$	$M_{cd}(b), M_{sd}(b)$	$w_u(b)$	$w_d(b)$
14.4 Kbps	1	1	0.4	0.21
28.8 Kbps	1	1	0.4	0.21
33.6 Kbps	1	1	0.4	0.21
56 Kbps	1	1	0.4	0.21
64 Kbps	2	1	0.53	0.72
128 Kbps	2	1	0.53	0.72
DSL	2	2	0.53	0.72
Cable	2	2	0.53	0.72
T1	3	2	0.07	0.07
T3	3	2	0.07	0.07

$H_2$ service parameters	
$\mu_1$	0.001
$\mu_2$	0.1
$\alpha_1$	0.6
$\alpha_2$	0.4

Table 3: Model parameters used for the numerical experiments

model results in selected cases where theoretical results are known or can be exactly computed. In particular, we compared the model results with the ideal case where there is no competition for the bandwidth of the tagged peers and the tagged transfer is the only activity carried out by the two participants; in these cases we found a perfect agreement between the model predictions and the theoretical results.

All the results presented in this section have been computed on a 1.5 GHz Pentium IV personal computer equipped with 1 GB of main memory and running the Linux operating systems. Each solution of the stochastic flow model requires a few minutes (usually between two and three) of CPU time.

**The c\_peer bandwidth and the resource popularity** The first set of experiments we performed is to evaluate how the c\_peer bandwidth impacts on the performance indexes we defined. We considered three possible cases for the c\_peer bandwidth (56 Kbps modem, DSL and Cable), for increasing popularity of the considered resource, i.e.,  $n$  equal to 1, 50, and 200, and for two different user bandwidth distribution (“scenario n.1” and “scenario n.2”). Table 4 reports the values of the average transfer time, the 50<sup>th</sup> and the 90<sup>th</sup> quantiles, for the two scenarios, for different c\_peer bandwidths, and different values of popularity. We considered a resource whose size  $s$  is equal to 4 MB.

The case  $n = 1$  considers a not-widespread resource; in this case, the c\_peer performs a random choice when selecting a s\_peer using the probability distribution of Table 1 and 2 for the scenario n. 1 and n. 2 respectively.

For  $n = 50$  and  $n = 200$  the c\_peer selects a s\_peer according to the distribution obtained by using the recurrence Equation (11). Due to the s\_peer selection policy we consider, increasing the resource popularity makes the probability of selection a high bandwidth s\_peer close to 1. In turn, this makes the c\_peer bandwidth the bottleneck for the resource transfer.

For a c\_peer with a slow connection (56 Kbps modem) the bottleneck is the c\_peer bandwidth

Performance Index	n	scenario n. 1			scenario n. 2		
		56 Kbps (sec)	DSL (sec)	Cable (sec)	56 Kbps (sec)	DSL (sec)	Cable (sec)
average	1	1414.2	773.1	773.4	1160.9	258.7	258.3
50 <sup>th</sup> quantile		1215.0	120.0	122.0	1119.0	92.0	92.0
90 <sup>th</sup> quantile		2109.0	2108.0	2109.0	1431.0	421.0	421.0
average	50	1114.0	68.6	77.3	1113.9	68.6	77.3
50 <sup>th</sup> quantile		1104.0	56.0	71.0	1104.0	56.0	71.0
90 <sup>th</sup> quantile		1348.0	110.0	110.0	1348.0	110.0	110.0
average	200	1114.0	50.9	64.4	1113.9	50.9	64.3
50 <sup>th</sup> quantile		1104.0	50.0	63.0	1104.0	50.0	63.0
90 <sup>th</sup> quantile		1348.0	61.0	78.0	1348.0	61.0	78.0

Table 4: Model results for the two scenarios, for different c\_peer bandwidths, and different values of resource popularity.

and hence the s\_peer selection policy is not a critical issue. In particular we can see that the difference between the average transfer time for  $n = 200$  drops by 20% with respect to the case  $n = 1$  (that also corresponds to a random s\_peer selection policy).

On the other hand in the case of a DSL c\_peer the case  $n = 200$  makes the average transfer time drop by  $\sim 90\%$  with respect to the case  $n = 1$ . The same holds for the CABLE connection. From these considerations we can conclude that the effect of the resource popularity is much more evident in case of c\_peer with high speed bandwidth. Another way of looking these results is that the impact of the s\_peer selection policy is stronger in case of c\_peers with high speed bandwidth.

From the comparison between the two scenarios we can observe that, due to the s\_peer selection policy we consider, the differences between the two scenarios can be appreciated only for not-widespread resource. Furthermore we can see that the differences between the measures obtained for the two scenarios also depend on the c\_peer bandwidth connection. In particular, for the scenario n. 2 and  $n = 1$  the average transfer time drops by 18% for the 56 Kbps modem c\_peer and by  $\sim 65\%$  for a DSL or a CABLE c\_peer.

**The P2P network load** The results we presented in Table 4 have been obtained by using a value for  $\lambda_0$  equal to 7.0 (this value is derived from the measures reported in [23]). We studied the model as function of the load of the P2P network varying the value of  $\lambda_0$ . We started from a maximum number equal to  $\lambda_0 = 7.0$  and we considered four (lower) additional values: 2.0, 0.5, 0.125 and 0.0625. The results reported in Table 5 refer to a file transfer whose size  $s$  is equal to 4 MB and the number  $n$  of available resources is equal to 5.

Since the load of the P2P network corresponds to the number of concurrent peers that share the available s\_peer bandwidth during the file transfer of the tagged c\_peer, when the P2P network load increases the time needed to the file transfer increases. If we look at the

Performance Index	$cb$	scenario n.1				
		$\lambda_0$				
		7.0	2.0	0.5	0.125	0.0625
average	56 Kbps	1116.3	1112.6	1099.1	1055.2	1011.4
50 <sup>th</sup> quantile		1105.0	1102.0	1088.0	1047.0	1007.0
90 <sup>th</sup> quantile		1351.0	1374.0	1332.0	1294.0	1266.0
average	DSL	96.5	96.0	94.7	91.2	87.8
50 <sup>th</sup> quantile		86.0	86.0	85.0	82.0	78.0
90 <sup>th</sup> quantile		112.0	111.0	109.0	105.0	102.0

Performance Index	$cb$	scenario n.2				
		$\lambda_0$				
		7.0	2.0	0.5	0.125	0.0625
average	56 Kbps	1114.2	1110.6	1097.0	1053.0	1009.1
50 <sup>th</sup> quantile		1105.0	1101.0	1088.0	1046.0	1007.0
90 <sup>th</sup> quantile		1348.0	1344.0	1329.0	1291.0	1263.0
average	DSL	85.5	85.0	83.7	80.2	76.9
50 <sup>th</sup> quantile		85.0	85.0	83.0	80.0	77.0
90 <sup>th</sup> quantile		109.0	108.0	106.0	102.0	99.0

Table 5: Model results for the two scenarios and for different P2P network loads.

values of the average, we can note that, for both scenarios, when the overall load increases from 0.0625 to 7.0 (more than 100 times higher) the increase is  $\sim 10\%$  both for the 56 Kbps modem and for the DSL. It should also be noted that the increase is in any case small because the model does not account for the queuing time before starting the file transfer.

**The user behavior** This analysis regards the user behavior, and in particular the concepts of participation and incentives in P2P networks. In [2] the user traffic in Gnutella is analyzed leading to conclude that there is a significant amount of free riding in the system. Free riding is the phenomenon where the peers only consume (download) and stop producing (upload); it is well known that this peers behavior can degrade considerably the system performance.

To motivate peers to be cooperative in P2P applications it could be useful to introduce some incentives policy. The Kazaa protocol uses a participation level that characterize each peer according to its behavior in the P2P community. The higher is the participation level the higher is the priority of the peer when it has to enqueue in a s\_peer buffer for a resource transfer. Other examples of incentive are proposed, see for instance [35]. In our work we assume an incentive policy in order to demonstrate that the good behavior of the peers can improve the performance of the whole system. We defined a policy that provides the c\_peer an extra 10% of available s\_peer bandwidth (obviously limited to the maximum) for each allowed upload.

Performance Index	$cb - sb$	$M_{cu}(cb)$			
		0	1	2	3
average	CABLE - CABLE	196.2	178.6	163.8	196.3
50 <sup>th</sup> quantile	CABLE - CABLE	195.0	177.0	162.0	195.0
90 <sup>th</sup> quantile	CABLE - CABLE	237.0	216.0	198.0	237.0

Table 6: Results for different number of maximum uploads allowed by the c\_peer.

We assumed that this rule is valid for the tagged transfer only and we calculated the file transfer time distribution for a CABLE c\_peer downloading from a CABLE s\_peer a 4 MB file as a function of the maximum number of uploads allowed by the c\_peer ( $M_{cu}$ ).

The results reported in Table 6 show that as consequence of the incentives policy the c\_peer can improve its performance and let the number of available resource in the whole system increase. Indeed when the c\_peer allows 1 and 2 maximum uploads the performance improves respect to the case with no uploads allowed, but when it allows 3 uploads the performance are very close to the ones obtained without cooperation, i.e. when maximum number of uploads is equal to 0. We conclude that a threshold exists on the maximum number of allowed uploads above which improvements guaranteed by the extra bandwidth reservation are offset by the increased uploads request by other peers.

These results refer to a  $\lambda_0$  value equal to 2.0 but we verified the same behavior for every value of  $\lambda_0$  considered in the load experiments of Table 5.

## 7 Conclusions

In this paper we propose a stochastic fluid flow model to analyze the transfer time distribution for the download of a file in a P2P network. The fluid level is used to represent the amount of bytes downloaded by a user participating in a P2P file sharing application; the flow rate is modulated by a set of discrete states representing the concurrent upload and download operations on both the c\_peer and the s\_peer chosen for the download. A transient solution of the model is then performed to compute the transfer time distribution for a file of a given size.

The model is developed to account for several aspects related to the file transfer in P2P networks; in particular, the impact of file popularity, s\_peer selection policy, bandwidth characteristics, concurrent downloads and uploads, cooperation level of both c\_peers and s\_peers are included in our model specification.

The model has been used to investigate the impact of several system parameters on interesting quantities that can be derived from the transfer time distribution. In particular, we investigated the impact of c\_peer bandwidth, and overall interfering load on the s\_peer for two different distributions of the peer bandwidth. The analysis confirmed our intuition that increasing interfering load yield worse user performance in terms of average and percentiles of



the completion time distribution; furthermore, we verified that the bandwidth of the c\_peer is the network bottleneck when the file popularity is high and the selection policy favors peers with large bandwidth.

A more interesting investigation we performed is on the effect of cooperation incentives on the transfer time distribution; we assumed that a fraction of the bandwidth of the s\_peer is reserved to the tagged c\_peer that is cooperating and we evaluated the effect of incentives by increasing the maximum number of concurrent uploads allowed by peers. We found that a threshold exists on the maximum number of allowed uploads above which improvements guaranteed by the extra bandwidth reservation are offset by the increased uploads request by other peers.

The model can be extended in several ways and a general framework to include parallel downloads, hot-spot phenomena on popular peers, and resource distribution has been outlined; other extensions we are currently working on include the behavior of s\_peers that queue uploads before serving requests and peer availability issues to consider users that alternate between connection and disconnection to the P2P network.

Furthermore, we are currently involved in the design and implementation of a fluid-based simulation approach for peer-to-peer networks to provide a scalable framework to analyze large size systems as well as to validate analytical models.

## References

- [1] Top applications (bytes) for subinterface: Sd-nap traffic. Technical report, CAIDA workload analysis of SD-NAP data, 2002. URL <http://www.caida.org/analysis/workload/byapplication/dsnap/index.xml>.
- [2] E. Adar and B. Huberman. Free Riding in Gnutella. *First Monday*, 5(10), Oct 2000. [http://www.firstmonday.dk/issues/issue5\\_10/adar/](http://www.firstmonday.dk/issues/issue5_10/adar/).
- [3] D Anick, D. Mitra, and M. M. Sondhi. Stochastic Theory of a Data-Handling System. *Bell Sys. Thech. J*, 61(8):1871–1894, Oct 1982.
- [4] F. Clevenot and P. Nain. A Simple Model for the Analysis of SQUIRREL. In *Proceedings of INFOCOM 2004*, Hong Kong, Mar 2004.
- [5] F. Dabek, E. Brunskill, M. F. Kaashoek, D. Karger, R. Morris, I. Stoica, and H. Balakrishnan. Building Peer-to- Peer Systems with Chord, a Distributed Lookup Service. In *Proceedings of the 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII)*, Schloss Elmau, Germany, 2001.
- [6] A. I. Elwalid and D. Mitra. Statistical Multiplexing with Loss Priorities in Rate-Based Congestion Control of High-Speed Networks. *IEEE Transaction on Communications*, 42(11):2989–3002, November 1994.

- [7] 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.
- [8] R. Gaeta, M. Gribaudo, D. Manini, and M. Sereno. Fluid Stochastic Petri Nets for Computing Transfer Time Distributions in Peer-to-Peer File Sharing Applications. Technical report, Dipartimento di Informatica, Università di Torino, Torino, Italia, 2004. URL: [http://www.di.unito.it/~matteo/TR\\_GMMS04.pdf](http://www.di.unito.it/~matteo/TR_GMMS04.pdf), Submitted for publication.
- [9] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley. Modeling Peer-Peer File Sharing System. In *Proceedings of INFOCOM 2003*, San Francisco, USA, Apr 2003.
- [10] R. German, M. Gribaudo, G. Horvath, and M. Telek. Stationary Analysis of FSPNs with mutually dependent Discrete And Continuous Part. In *Proc. 10th Int. Workshop on Petri Nets and Performance Models (PNPM'03)*, Urbana, Illinois, USA, September 2003. IEEE Comp. Soc. Press.
- [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] G. Horton, V. G. Kulkarni, D. M. Nicol, and K. S. Trivedi. Fluid stochastic Petri Nets: Theory, Application, and Solution Techniques. *European Journal of Operations Research*, 105(1):184–201, Feb 1998.
- [13] G. Kan. Chapter 8: Gnutella. In A. Oram, editor, *Peer-to-Peer Harnessing the Power of Disruptive Technologies*. O'Reilly, 2001.
- [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] A. Kapur, N. G. Harold, I. Marcus, R. Brooks, and S. Rai. Performance and Design of P2P Networks for Efficient File Sharing. In *Industrial Engineering Research Conference*, Portland, Oregon, USA, 2003.
- [17] L. Kosten. Stochastic theory of data-handling system with groups of multiple sources. In *Proceedings IFIP WG 7.3/TC 6 Second International Symposium of the Performance of Computer-Communication Systems*, Zurich, Switzerland, 1984.

- [18] V. Misra, W. B. Gong, and Towsley. Stochastic Differential Equation Modeling and Analysis of TCP Window Size Behavior. In *Proc. Performance 1999*, Istanbul, Turkey, 1999.
- [19] V. Misra, W. B. Gong, and Towsley. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an application to RED. In *Proc. SIGCOMM '2000*, Stockholm, Sweden, 2000.
- [20] D. Mitra. Stochastic Theory of a Fluid Model of Producers and Consumers. *Adv. Appl. Prob.*, 20:646–676, 1988.
- [21] E. Ng, Y. hua Chu, S. G. Rao, K. Sripanidkulchai, and H. Zang. Measurement-based optimization techniques for bandwidth-demanding peer-to-peer systems. In *Proceedings of INFOCOM 2003*, San Francisco, USA, Apr 2003.
- [22] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content-Addressable Network. In *Proceedings of ACM SIGCOMM '01*, Aug 2001.
- [23] 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.
- [24] M. Ripeanu, I. Foster, and A. Iamnitchi. Mapping the Gnutella Network: Properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Computing*, 6(1):50–57, jan-feb 2002.
- [25] J. Ritter. Why Gnutella can't scale no really. Technical report, URL <http://www.tch.org/gnutella.html>, 2001.
- [26] A. I. T. Rowstron and P. Drushel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware*, 2001.
- [27] 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.
- [28] S. Saroiu, P. K. Gummadi, and S. D. Gribble. A Masurement Study of Peer-to-Peer File Sharing System. In *In Proceedings Of Multimedia Computing and Networking (MMCN) 2002*, Jan 2002.
- [29] 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.

- [30] B. Sericola. Transient Analysis of Stochastic Fluid Models. *Performance Evaluation*, 32(4):245–263, 1998.
- [31] T. Tanaka, O. Hashida, and Y. Takahashi. Transient Analysis of Fluid Models for ATM Statistical Multiplexer. *Performance Evaluation*, 23:145–162, 1995.
- [32] K. Trivedi, D. Chen, and Y. Hong. Second Order Stochastic Fluid Flow Models with Fluid Dependent Flow rates. *Performance Evaluation*, 49:341–358, 2002.
- [33] B. Yang and H. Garcia-Molina. Comparing Hybrid Peer-to-Peer Systems. In *7th International Conference on Very Large Data Bases (VLDB01)*, Rome, Italy, 2001.
- [34] X. Yang and G. de Veciana. Service Capacity of Peer to Peer Networks. In *Proceedings of INFOCOM 2004*, Hong Kong, Mar 2004.
- [35] B. Yu and M. P. Singh. Incentive Mechanisms for Peer-to-Peer Systems. In *In Proceedings of the Second International Workshop on Agents and Peer-to-Peer Computing*, 2003.

## Appendix: Model Construction

In this section we summarize all the steps required to produce the solution of the proposed model on a simple example. In particular we will consider the case in which the c\_peer uses a 56 kbps modem. The procedure will require several iterations, one for each possible s\_peer bandwidth. We will consider in detail only the iteration for a specific bandwidth (i.e. 64 kbps). The other bandwidths can be evaluated in a similar way.

1. Consider a possible s\_peer bandwidth.
2. Generate a fluid model for the particular pair of values for  $cb$  and  $sb$ . In particular the algorithm builds the four components ( $cu$ ,  $cd$ ,  $su$  and  $sd$ ) and defines their state spaces. Consider for example also that the maximum number of upload is  $K_{cu} = K_{su} = 2$  and the maximum number of download is  $K_{cd} = K_{sd} = 1$ . The state spaces of the four components are:

$$\begin{aligned}
 \mathcal{S}_{cu} &= \{s_{0:}, s_{1:h}, s_{1:l}, s_{2:hh}, s_{2:hl}, s_{2:ll}\} \\
 \mathcal{S}_{cd} &= \{s_{0:}\} \\
 \mathcal{S}_{su} &= \{s_{0:}, s_{1:h}, s_{1:l}\} \\
 \mathcal{S}_{sd} &= \{s_{0:}, s_{1:h}, s_{1:l}\}.
 \end{aligned}$$

The notation for the state labels is:  $s_{n:d}$ , where  $n$  represents the number of costumers in the queue, and  $d$  is the description of how the costumers are distributed among the various hyper-exponential stages. The two stages of the hyper-exponential are identified with  $h$  and  $l$ . In this way we have that:  $s_{0:}$  represents the state of an empty queue,  $s_{1:h}$  one costumer in stage  $h$ ,  $s_{1:l}$  one costumer in stage  $l$ ,  $s_{2:hh}$  two costumers in stage  $h$ ,

$s_{2:hl}$  one costumer in stage  $h$  and one costumer in stage  $l$ ,  $s_{2:ll}$  two costumers in stage  $l$ . Note that since there is at least the tagged transfer, we have that the maximum length of the queue in  $\mathcal{S}_{cd}$  is zero even if  $K_{cd} = 1$ , and that the maximum length in  $\mathcal{S}_{su}$  is one even if  $K_{su} = 2$ .

3. For the considered s\_ppeer bandwidth, construct matrices  $\mathbf{Q}_{cu}$ ,  $\mathbf{Q}_{cd}$ ,  $\mathbf{Q}_{su}$  and  $\mathbf{Q}_{sd}$ . For example:

$$\mathbf{Q}_{cu} = \begin{vmatrix} -\lambda_{cu}(cb) & \lambda_{cu}(cb)\alpha_1 & \lambda_{cu}(cb)\alpha_2 & 0 & 0 & 0 \\ \mu_1 & -\lambda_{cu}(cb) - \mu_1 & 0 & \lambda_{cu}(cb)\alpha_1 & \lambda_{cu}(cb)\alpha_2 & 0 \\ \mu_2 & 0 & -\lambda_{cu}(cb) - \mu_2 & 0 & \lambda_{cu}(cb)\alpha_1 & \lambda_{cu}(cb)\alpha_2 \\ 0 & 2\mu_1 & 0 & -2\mu_1 & 0 & 0 \\ 0 & \mu_1 & \mu_2 & 0 & -\mu_1 - \mu_2 & 0 \\ 0 & 0 & 2\mu_2 & 0 & 0 & -2\mu_2 \end{vmatrix} \quad \mathbf{Q}_{cd} = \begin{vmatrix} 0 \end{vmatrix}$$

$$\mathbf{Q}_{su} = \begin{vmatrix} -\lambda_{su}(sb) & \lambda_{su}(sb)\alpha_1 & \lambda_{su}(sb)\alpha_2 \\ \mu_1 & -\mu_1 & 0 \\ \mu_2 & 0 & -\mu_2 \end{vmatrix} \quad \mathbf{Q}_{sd} = \begin{vmatrix} -\lambda_{sd}(sb) & \lambda_{sd}(sb)\alpha_1 & \lambda_{sd}(sb)\alpha_2 \\ \mu_1 & -\mu_1 & 0 \\ \mu_2 & 0 & -\mu_2 \end{vmatrix}$$

4. Combine the previous states spaces with cartesian product to obtain the complete state space:  $\mathcal{S} = \mathcal{S}_{cu} \times \mathcal{S}_{cd} \times \mathcal{S}_{su} \times \mathcal{S}_{sd}$ . In the example iteration, the resulting process will have  $6 * 1 * 3 * 3 = 54$  states. In particular they will be:

$$\mathcal{S}_{cu} = \{(s_{0:}, s_{0:}, s_{0:}, s_{0:}), (s_{0:}, s_{0:}, s_{0:}, s_{1:h}), (s_{0:}, s_{0:}, s_{0:}, s_{1:l}), (s_{0:}, s_{0:}, s_{1:l}, s_{0:}), \dots, (s_{2:ll}, s_{0:}, s_{1:l}, s_{1:l}), (s_{2:hl}, s_{0:}, s_{1:l}, s_{1:h}), (s_{2:ll}, s_{0:}, s_{1:l}, s_{1:l})\}$$

5. Combine the previous matrices using the Kronecker's sum to obtain the infinitesimal generator of the stochastic fluid model:  $\mathbf{Q} = \mathbf{Q}_{cu} \oplus \mathbf{Q}_{cd} \oplus \mathbf{Q}_{su} \oplus \mathbf{Q}_{sd}$ .
6. Compute matrix  $\mathbf{R}$  evaluating for each state the corresponding transfer rate using functions  $f_s(s_{su}, s_{sd})$  and  $f_c(s_{cu}, s_{cd})$ . In the example iteration we have:

$$\mathbf{R} = \begin{vmatrix} \min(f_s(s_{0:}, s_{0:}), f_c(s_{0:}, s_{0:})) & 0 & \dots & 0 \\ 0 & \min(f_s(s_{0:}, s_{0:}), f_c(s_{0:}, s_{1:h})) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \min(f_s(s_{2:ll}, s_{0:}), f_c(s_{1:l}, s_{1:l})) \end{vmatrix}$$

7. Compute the initial distribution for each of the four components by solving the following systems of equations:

$$\begin{cases} \pi_{cu}^0 \mathbf{Q}_{cu} = 0 \\ \pi_{cu}^0 \mathbf{1}_{cu} = 1 \end{cases} \quad \begin{cases} \pi_{cd}^0 \mathbf{Q}_{cd} = 0 \\ \pi_{cd}^0 \mathbf{1}_{cd} = 1 \end{cases} \quad \begin{cases} \pi_{su}^0 \mathbf{Q}_{su} = 0 \\ \pi_{su}^0 \mathbf{1}_{su} = 1 \end{cases} \quad \begin{cases} \pi_{sd}^0 \mathbf{Q}_{sd} = 0 \\ \pi_{sd}^0 \mathbf{1}_{sd} = 1 \end{cases}$$

8. Determine the initial probability of the fluid model using the Kronecker's product:  $\pi_0 = \pi_{cu}^0 \otimes \pi_{cd}^0 \otimes \pi_{su}^0 \otimes \pi_{sd}^0$ .

9. Solve the following system of partial differential equations that computes the transient solution of the fluid model:

$$\left\{ \begin{array}{l} \frac{\partial \boldsymbol{\pi}(\tau, x)}{\partial \tau} + \frac{\partial \boldsymbol{\pi}(\tau, x)}{\partial x} \mathbf{R} = \boldsymbol{\pi} \mathbf{Q} \\ \boldsymbol{\pi}(0, x) = \delta(x) \boldsymbol{\pi}_0, \\ \boldsymbol{\pi}(\tau, 0) = \mathbf{0}. \end{array} \right.$$

The previous system can be solved using any suitable solution, as outlined in Section 3

10. Determine the transfer time distribution for the considered s\_peer bandwidth, computing the following integral:

$$F_t(t|sb, s) = \int_s^\infty \boldsymbol{\pi}(\tau, x) \mathbf{1} dx \Big|_{\tau=t}$$

11. Repeat steps 2-10 for every possible s\_peer bandwidth  $sb$ .
12. Compute the final distribution, weighting each s\_peer bandwidth specific distribution with a coefficient dependent on the popularity of the file:

$$F_t(t|cb, s, n) = \sum_{sb} F_t(t|sb, s) P(SB = sb | N = n).$$

In all the previous steps the dependency on the c\_peer bandwidth  $cb$  as been omitted to simplify the formulas.