Modelling Search, Availability, and Parallel Download in P2P File Sharing Applications with Fluid Model

Daniele Manini, Marco Gribaudo
Dipartimento di Informatica, Universitá di Torino
C.so Svizzera 185, Torino, Italia, {manini,marcog}@di.unito.it

Abstract

ile transfer using Peer-to-Peer file sharing applications is usually divided into two steps: resource search and resource download. Depending on the file size and its popularity, either of the two phases can become the bottleneck. In this paper we describe both the location and download phases of a generic Peer-to-Peer file sharing application using a fluid model. The proposed model allows the computation of the transfer time distribution, and it is capable of considering some advanced characteristic such as parallel downloads and on-off peer behavior. Model parameters reflect network, application, resource and user characteristics, and can be tuned to analyze a large number of different real Peer-to-Peer implementations.ile transfer using Peer-to-Peer file sharing applications is usually divided into two steps: resource search and resource download. Depending on the file size and its popularity, either of the two phases can become the bottleneck. In this paper we describe both the location and download phases of a generic Peer-to-Peer file sharing application using a fluid model. The proposed model allows the computation of the transfer time distribution, and it is capable of considering some advanced characteristic such as parallel downloads and on-off peer behavior. Model parameters reflect network, application, resource and user characteristics, and can be tuned to analyze a large number of different real Peer-to-Peer implementations.F

1. Introduction

In Peer-to-peer (P2P) file sharing applications, the Quality of Service (QoS) perceived by users strongly depends on the time spent for locating the resource and on the time required to download it. In this paper we extend the fluid modeling technique presented in [6], [19] for the evaluation of the transfer time distribution in P2P file sharing systems.

We derive QoS user-perceived measures related to the searching and transfer phases for a given resource. With respect to [6], [19], we provide a method for the estimation of transfer time distribution in a P2P system adding advanced features such as the search phase, queueing time, the unavailability of the uploader peer, and the parallel download from multiple sources. Thanks to these additions we are able to include interesting issues related to both peers behavior (e.g., unavailability of the resource due to the peer leaving) and overlay topology (e.g., searching and queueing phases, and multiple downloads).

[0]1-4244-0715-X/06/$20.00©2006 IEEE
IEEE Catalog Number: 06EX1537;
ISBN: 1-4244-0715-X;
Library of Congress: 2006934023

2. Related Works

In this section we relate our model to similar approaches proposed in the literature. The model developed in [7] (one of the firsts mathematical P2P models) explores three different type of architecture (centralized indexing, distributed indexing with flooded queries, and distributed indexing with hashing directed queries). It allows to analyze system scalability, free riding problem, and resource issues such as popularity and availability. An analytical model based on queueing networks has been presented in [18], this work is able to capture both network and overlay characteristics. The study presented in [21] addresses the problem of optimal server peer selection for both P2P downloading and P2P streaming. The problem has been studied within the framework of a free-market resource economy in which peers buy and sell resources directly from each other.

The work in [2] 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. The resulting stochastic fluid model turns out to be mathematically tractable, and provides a simple and low-complexity procedure for computing the hit probability.
The work presented in [20] shows a simple fluid model for BitTorrent P2P application to study steady-state performance measures, such as average number of downloaders and average download time as a function of several parameters (e.g. peer arrival rate, peer leaving rates and uploading bandwidth). The proposed model allows to investigate scalability, file sharing efficiency, system stability, and incentives to prevent free-

TABLE 1: MODEL NOTATIONS

Notation	Description	Range
$\mathcal{B}$	Set of bandwidths	{14.4, 28.8, 33.6, 56, 64 128, DSL, Cable, T1, T3}
SB	Server bandwidth	$\mathcal{B}$
CB	Client bandwidth	$\mathcal{B}$
S	Resource size	$\mathbb{N}$
$K(b)$	Max. number of concurrent peers	$\mathcal{B} \rightarrow \mathbb{N}$
$L(b)$	Average number of requests of uploads	$\mathcal{B} \rightarrow \mathbb{R}$

riding.

Our work extends the results presented in [6], [19]. In particular, we consider the possibility that a server can leave the system before the transfer is completed. We model also the following search phase, including the time spent in queue, for a new source. Furthermore we describe the transfer of a resource with multiple downloads, that is a functionality performed by all P2P file sharing applications (e.g., eDonkey, BitTorrent, etc.). This study allows us to investigate the user perceived QoS in a more realistic scenario and to better understand P2P systems.

Extensive validation of results for our modeling technique faces the same difficulties as most of the previous works based on analytical models. Simulation frameworks presented in literature, such as [13], [11] just to mention few, do not allow us to validate our model results since they do not provide transfer time information. However we perform a simple safety check shown in Section 4.

3. PEER-TO-PEER MODEL

We propose a fluid model for the estimation of the transfer time distribution in P2P file sharing applications. The model will be described using the Fluid Stochastic Petri Net (FSPN) formalism [12], [9]. This section will be structured as follows: in Section 3-A we will summarize the original model described in [19]. Following that notation, we call *client* the downloader and *servers* the uploaders. Table 1 reports the other notations derived from the reference. We will then present the novel extensions proposed by this work: in Section 3-B we will consider the search, queueing and on-off behavior of servers, and in Section 3-C we will introduce the parallel download from multiple servers.

A. The FSPN Model

The basic model [19] computes the transfer time distribution of a resource of size S downloaded by a client with a bandwidth CB, from a server with bandwidth SB. The server simultaneously uploads other concurrent peers. The basic model neglects both the search and the queueing phase, and download interruptions. That model is defined by means of a FSPN. This formalism adds fluid (continuous) variables to Stochastic Petri Nets [1], by introducing a new primitive called "Fluid Place". In a FSPN the portion of a model described using only ordinary Petri Nets primitives (i.e. discrete Places and Transitions) is addressed as the *discrete part of the model*. The state of the discrete part is used to modulate the flow rate, and it determines the evolution of the fluid part (i.e. the value of the continuous variable).

In the P2P model, the discrete part represents the load of the server. A fluid place is used to represent the amount of bytes transferred by the client. The main assumption in the basic model is that the session time of concurrent peers is described by an Hyperexponential distribution (with parameters α, μ_1 and μ_2), and that the interarrival time of concurrent downloaders is approximated by an exponential distribution (whose parameter $L(sb)$ is bandwidth dependent). The maximum number of concurrent downloads allowed on the server is limited by a bandwidth dependent parameter $K(sb)$. Moreover, the server bandwidth is equally shared among the concurrent downloaders. For a discussion on the validity of these assumptions, please refer to [19].

Using these assumptions, the available bandwidth at the client can be computed as a function of the number of concurrent peers. In particular, if we call I_j the total number of concurrent peers in a discrete state of FSPN model, then the available bandwidth is equal to:

$$f(I_j) = \min\left(\frac{sb}{I_j + 1}, cb\right) \quad . \tag{1}$$

Note that plus 1 takes into account the client itself. The FSPN model is analyzed by solving the system of partial differential equations that describes its underlaying stochastic process. From the solution to these equations the probability density $\bar{\pi}(\tau, x)$ of the fluid level at a given time instant τ can be directly computed. $\bar{\pi}(\tau, x)$ corresponds to the probability density that the number of bytes downloaded at time τ is equal to x. By integrating this quantity, the probability distribution that a file of size s can be downloaded in less than t time unit (i.e. the transfer time distribution) can be computed:

$$F_t(t|s) = \int_s^{\infty} \bar{\pi}(t, x)dx. \tag{2}$$

Please refer to [19] for a detailed description of the solution technique. In this work, we focus only on resource transfers between DSL bandwidth client and servers, however the analysis can be extended to any peer's bandwidth.

B. Modeling the search time, queueing time and peer unavailability

Search time is conditioned by many factors such as the popularity of the resource, protocol characteristics, the participation level of the user and the number of neighbor peers. After the searching phase the client selects peers from which get the resource. Queueing time is the time spent before a selected server serves the client request. It also depends on many factors, such as the maximum number of concurrent downloads allowed, the bandwidth of the server and the number of concurrent peers, the protocol, and the participation level of peers.

We simplify the model by considering *the aggregate search plus queuing time perceived by a client*. That is, we suppose that we could compute the distribution $QS(\tau)$ of the time required from the start of the search to the start of the actual download of a resource. This seems to be a quite strong

assumption, but in Section 4 we will prove that despite its simplicity, the proposed model is able to get most of the qualitative features that characterize parallel download in peer to peer applications.

Figure 1 represents the extension of the model proposed in [19]. The arrival of a new concurrent download is modelled by transition request_arrival whose rate is $L(sb)$. The session length distribution is modelled by the sub-net composed by places CHOICE, STAGE_1, STAGE_2, END_SERVICE and transitions choose_1, choose_2, terminate_service, service_1, service_2. Their parameters are directly mapped to the parameters of the distributions outlined in Section 3-A. The maximum number of concurrent downloads is determined by the initial marking of place AVAILABLE, and is set according to parameter $K(sb)$. The amount of byte transferred is modelled by fluid place TRANSFERRED and fluid transition transfer. The value of parameter I_j (total number of concurrent peers) corresponds to the sum of the marking of places STAGE_1 and STAGE_2. The search and queueing phases are represented by the generic firing time transition TON, with distribution ϕ_{on}.

Due to the active/non-active peer dynamics the server may become unavailable and then its service is stopped. When failures occur, the client starts a new search of the same resource, and then it continues the download (likely from another peer), after experiencing a new queueing time. The failure of server is represented in the model by generic firing time transition TOFF, with firing time distribution ϕ_{off}. Place SandQ represents the search and queueing phases, and place TRANS the resource transfer phase.

As reported in [19], special care should be used to compute the initial distribution of the number of concurrent peers at the server. Indeed, it is shown that the transfer time is affected by the initial state of the server especially in the case of small resource transfer.The initial state of the places representing the concurrent peers at the server should be determined at the time when the actual transfer starts, i.e. at the firing of transition TON. When transition TON fires, it should set the number of tokens in places AVAILABLE, STAGE_1 and STAGE_2 according to the initial distribution, determined following the technique proposed in [19]. The setting of the initial state is achieved by an appropriate set of immediate transitions, weighted according to the initial state distribution. In order to simplify Figure 1, this sub-net has been removed and has been represented by the gray arrow labeled with *Set Initial state*. Similarly, when the server experiences a failure, all the places of the sub-model representing its state must be emptied. This also can be achieved by an appropriate set of immediate transition, which has been represented in Figure 1 by the gray arrow labeled with *Clear state*.

In this model, the popularity of the resource is considered when determining the rate of transition TON. A very popular resource will have a shorter search and queueing time, since will be available from more peers. A rare resource will instead have a very high searching and queueing time.

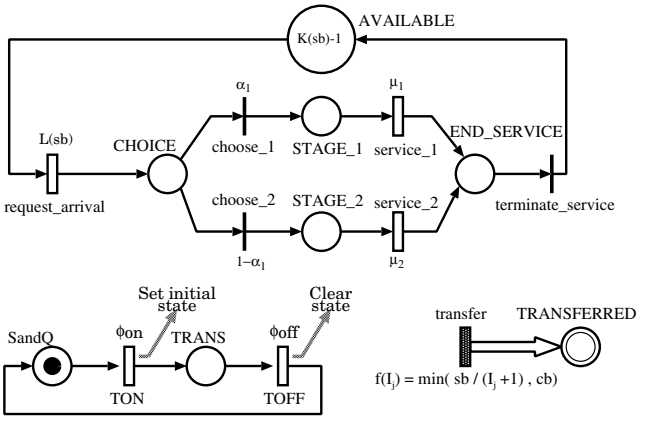

Fig. 1: FSPN model representation of an unreliable server with search and queueing phases.

C. Considering the parallel download from multiple sources

The model describing the parallel download from multiple servers can be obtained by repeating H times the sub-models of Figure 1 representing the server and the search-queueing state, where H corresponds to the maximum number of parallel downloads. This is shown in Figure 2. Note that the H sub-models representing the H servers, share the same resource download buffer, modeled by fluid place TRANSFERRED. In this case, the rate at which the file is downloaded, is expressed as the minimum between the client bandwidth cb, and the sum of the download rate from each server that is active in that time instant, that is:

$$f(I_j) = \min\left(\sum_{k=1}^{H} \mathcal{I}(\#_{\text{TRANS}_k} = 1)\frac{sb_k}{I_{jk}+1}, cb\right) \qquad (3)$$

where $\mathcal{I}(\#_{\text{TRANS}_k})$ is an indicator function that returns 1 if the the number of tokens in place TRANS of the submodel representing the k-th server is equal to 1 (i.e. active download), zero otherwise. I_{jk} represents the sum of the tokens in places STAGE_1 and STAGE_2 for each tangible (discrete) state $\boldsymbol{m}_j$ of the k-th server, i.e. the number of requests that interfere on the k-th server with the client service. As mentioned in Section 3-A plus 1 takes into account the client.

Despite the symmetries, the sub-models are not independent, since they are coupled by the fluid buffer TRANSFERRED. Moreover the relation that governs the rate of the growth of the fluid place (Equation 3) is non-linear, due to the presence of the $\min(\cdot)$ function. This prevents to apply a solution technique that analyzes each server separately, and combine them afterward.

4. EXPERIMENTS

In this section, we exploit the models presented in Section 3-B and Section 3-C to show how, despite their simplifying assumptions, they can describe the qualitative behavior of real peer to peer systems. In both cases, we present our analysis for the case when all peers have the same bandwidth connection (in particular, we consider 640 Kb/s DSL technology). We

TABLE 2: MODEL PARAMETERS USED FOR EXPERIMENTS

Service parameters	
μ_1	0.001
μ_2	0.1
α_1	0.6
α_2	0.4
Arrival rate	
L(DSL)	0.01

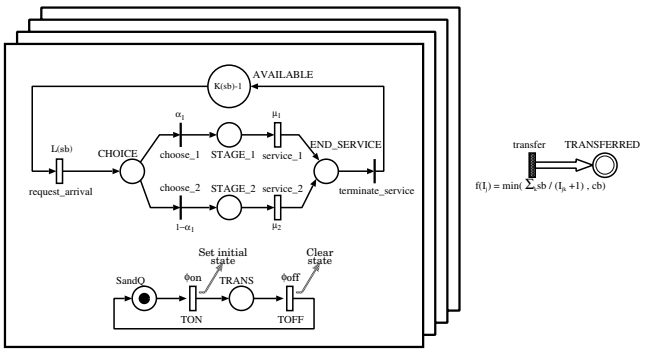

Fig. 2: FSPN model for multiple servers download.

approximate both search and queue time distribution and the server failure distribution with exponential distributions. In the following we will use parameters ϕ_{on} and ϕ_{off} to indicate the rate of the corresponding exponential distribution.

Extensive validation of results for our modeling technique shares the same difficulty of previous works on analytical models for P2P systems. It is a difficult task since existing measurement based studies 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 it would have a prohibitive programming and computational cost. Nevertheless, we performed simple validations by comparing model results in selected cases where theoretical results are known or can be exactly computed. In particular, we compared model results with the ideal case where there is no competition for the server bandwidth and the transfer is only conditioned by the minimum bandwidth between server and client. In these cases we found a perfect agreement between the model predictions and the theoretical results. It is a safety check that allow us to know that at least in the deterministic case, without concurrent operations, model result is identical to the expected one (that is the ratio between the resource size and the minimum bandwidth among client and server ones). Moreover, results presented in Table 4 are partially supported by the measurement study presented in [22]. In particular, in [22] it is shown that the average download speed is 30 KB/second that in the case of a $4MB$ resource corresponds to an average transfer time of 133 seconds. This average is comparable with most of average values shown in Table 4.

TABLE 3: DOWNLOADING BANDWIDTH VERSUS SESSION DURATION AND RESOURCE SIZE

Resource Size	Average Bandwidth (Kbit/sec)		
	Session Time 1000 sec	Session Time 20 sec.	Session Time 10 sec.
512 KB	24.32	10.4	6.56
4 MB	59.68	14.08	7.84
10 MB	68.48	6.64	3.12

First we evaluated the average bandwidth as function of the resource size (also considering different failure rates). We kept the searching-queueing rate constant to 0.01: this means that client wait a mean of 100 seconds to find a new connection. We vary the failure rate in order to get server sessions of 10, 20, and 1000 seconds. The maximum number of concurrent peers on the server, $K(sb)$, has been set to 4. In this analysis we does not consider parallel downloads. FSPN model parameters used in all experiments are reported in Table 2. The average bandwidth experienced to complete the transfer of the resource has been computed as the ratio between the resource size and the average of the time transfer. A first intuitive result (see Table 3) shows that the transfer time increases with the increasing of unavailability rate. However, we must point out that this effect heavily depends on the resource size. It is interesting to note that bigger resources suffer significantly from server failures. For instance, in the case of a 10 MBytes resource, the bandwidth falls down when the failure rate is 0.05 and 0.1 (that is session time of 10 and 20 seconds). Instead in the case of a 512 KByte resource, the penalty introduced by the failure of the server is less significative. This is due to fact that, on the average, the resource can be completely transferred before the server fails, despite shorter server session.

Most P2P file sharing applications (e.g., eDonkey, BitTorrent, etc.) allow parallel downloads. The model presented in Figure 2 represents this feature. We performed an experiment describing the transfer of a 4 MByte file. In this experiment searching-queueing rate has been set to 0.01 and failure rate has been set to 0.001, the maximum number of concurrent peers on each server $K(sb)$, has been set to 4. Client peer downloads from multiple sources and gets better performance when the number of source increases as shown in Fig. 3. However, improvements in performance are limited by the client download bandwidth; i.e. when the total bandwidth provided by multiple servers exceeds the maximum client download bandwidth, the speed at which the file is transferred remains constant, despite the growth in the number of sources. This is shown in table 4, where the mean and quantiles of the transfer time distribution related to a 4 MBytes resource are reported as function of the number of sources. In this experiment searching-queueing rate and failure rate has been set to 0.01 and 0.001 respectively, the maximum number of concurrent peers on each server, $K(sb)$, has been set to 2. It is possible to note that the improvement in transfer performance become less significative as the number of sources increases (since they saturate the client downloading bandwidth). Indeed when sources increase over 9 the time required to transfer the file remains constant. This insight may provide suggestions for the application design. E.g., if the application protocol is able to monitor the client bandwidth status it can detects when the

TABLE 4: TRANSFER TIME AS FUNCTION OF THE NUMBER OF SOURCES

Number of Sources	Tansfer Time (sec.)		
	Mean	50^{th} quantile	90^{th} quantile
3	178	170	240
4	148	140	200
5	131	130	170
6	119	120	160
7	110	110	148
8	104	100	130
9	104	100	130

client is the bottleneck, and then avoid to add new (parallel) sources. The contribute of refused sources, that should not be exploited for improving the client transfer performance, could be exploited to improve the system service capacity for other peers.

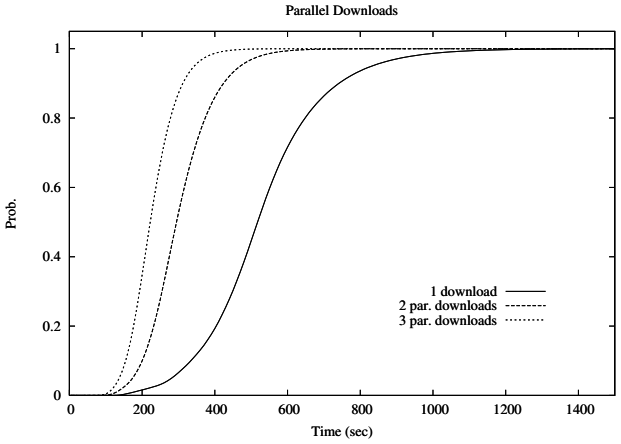

Fig. 3: Improvement provided by parallel downloads

Furthermore it is interesting to see how the benefit derived from the use of parallel download depends on the size of the resource. Consider the case in which downloading session does not suffer from the servers failures (i.e. the failure rate is very low), we set searching-queueing rate much bigger than the failure one respectively 0.1 and 0.001. The maximum number of concurrent peers on each server, $K(sb)$, has been set to 2. The study has been done for 512 KBytes, 4 and 10 MBytes resource sizes and for a number of parallel downloads that grows from 1 up to 6. As shown in Fig. 4 small resources take less benefits from parallel downloading, since the downloading time is shorter than the time required by the searching and queueing phase to start a parallel download from another source. For bigger resources instead, the downloading time is reduced significantly increasing the number of sources. These improvements are however limited by the client bandwidth, as shown in the previous experiment. This can be seen for for the 4 and the 10 MBytes cases, when the number of sources increases from 5 to 6.

In order to describe different system scenario we also approximated the searching and queueing rate with different distributions. Results presented so far are obtained by approximating rates with exponential distributions. In addition we modeled the searching and queueing phases with Hyper-

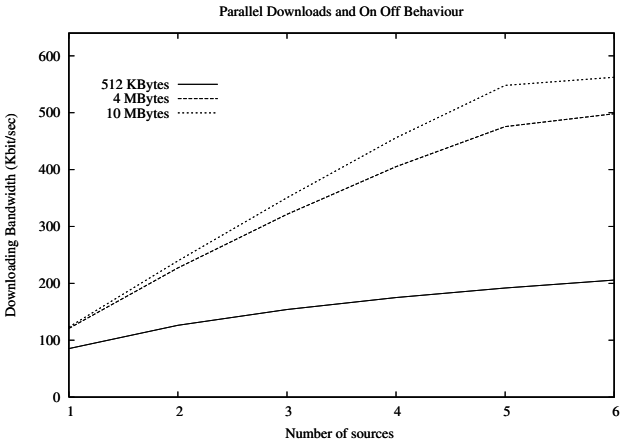

Fig. 4: Benefits of parallel downloading for different resource sizes

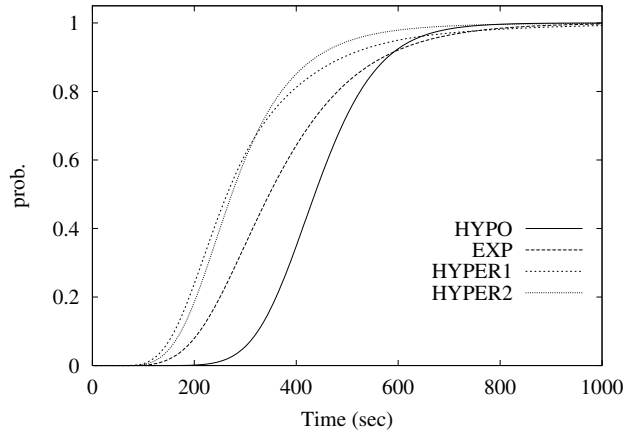

Fig. 5: Transfer time distribution with different searching-queueing rate distributions

exponential and Hypo-exponential distributions. Results reported in Table 5 and Figure 5 refers to the transfer of a $4MB$ file with 3 parallel downloads and a session mean time of 15 minutes. In all cases we set the mean time spent in the searching/queueing phase to 5 minutes. In order to get this value we set the parameters according to the different distribution we used. In the case "Hyper-exponential 1" the mean time spent by the client in the searching/queueing phase is 10 minutes with a probability of 44% and 1 minute with a probability of 56%. In this case faster searching/queueing phases are favorite, indeed transfer time is shorter than in the case "Exponential". Shorter searching/queueing phases are even more favorite in the case "Hyper-exponential 2"

TABLE 5: MODELING SEARCHING AND QUEUEING PHASES WITH DIFFERENT DISTRIBUTIONS.

Distribution	Tansfer Time (sec.)		
	Mean	90^{th} quantile	95^{th} quantile
Hypo-exponential	448	580	630
Exponential	376.31	575	660
Hyper-exponential 1	306.34	495	605
Hyper-exponential 2	297.35	440	510

where the mean time spent by the client is 3.45 minutes with probability 80%, and 10 minutes with probability 20%. This setting results in faster transfers, as reported in Table 5.

The choice of the Hyper-exponential distribution can be useful for describing different scenario where shorter searching/queueing phases model popular resource transfers, whereas longer ones model rare resource transfers. The Hypo-exponential distribution can be used to model rates when the approximation should be more deterministic. In this case, the Hypo-exponential case corresponds to a 5 stages Erlang distribution.

5. CONCLUSIONS

The Quality of Service perceived by peers strongly depends on the time spent for locating the resource and on the time required to get it. We extended the analytical modeling technique presented in [19] for computing the transfer time distribution, including searching, queueing and downloading phases. The model accounts for network, overlay application, and user characteristics. We derive results considering multiple downloads and on/off peer activities. The proposed model, is thus capable to capture many of the features that characterize current P2P file sharing applications.

ACKNOWLEDGMENTS

This work was partially supported by the Italian Ministry for Education, University and Research (MIUR) in the framework of the PRIN Famous and FIRB WebMinds projects.

REFERENCES

[1] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli and G. Franceschinis. Modelling with Generalized Stochastic Petri Nets. *John Wiley & Sons*, 1995.

[2] F. Clevenot and P. Nain. A Simple Model for the Analysis of SQUIRREL. In *Proceedings of INFOCOM 2004*, Hong Kong, Mar 2004.

[3] D. R. Cox. The Analysis of Non-Markovian Stochastic Processes by the Inclusion of Supplementary Variables. In *Proceedings of Cambride Philosophical Society*, 51:433–440, 1955.

[4] E. De Souza e Silva and R. Gail. An algorithm to calculate transient distributions of cummulative rate and impulse based reward. *Commun. in Statist. – Stochastic Models*, 14(3):509–536, 1998.

[5] L. Donatiello and V. Grassi. On evaluating the cumulative performance distribution of fault-tolerant computer systems. *IEEE Transactions on Computers*, 1991.

[6] R. Gaeta, M. Gribaudo, D. Manini, and M. Sereno. Analysis of Resource Transfers in Peer-to-Peer File Sharing Applications using Fluid Models. *Performance Evaluation: An International Journal*, Peer-to-Peer Computing Systems Vol. 63 Issue 3 Pages 147-264 (March 2006).

[7] 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.

[8] M. Gribaudo, M. Sereno, ans A. Bobbio. Fluid Stochastic Petri Nets: An Extended Formalism to Include non-Markovian Models Proc. 8^{th} Intern. Workshop on Petri Nets and Performance Models, Zaragoza, Spain, Spe 1999.

[9] 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.

[10] 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.

[11] 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 *11th Int. Sym. on Modeling, Analysis and Simulation of Computer Tel. System (MASCOT 2003)*, Orlando, Florida, 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] 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.

[14] H. Nabli and B. Sericola. Performability analysis: a new algorithm. *IEEE Transactions on Computers*, 45:491–494, 1996.

[15] S. Rácz and M. Telek. Numerical analysis of large Markovian reward models. Performance Evaluation, 36&37:95–114, 1999.

[16] 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.

[17] 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.

[18] K.K. Ramachandran, B. Sikdar. An Analytic Framework for Modeling Peer to Peer Networks. In *Proceedings of INFOCOM 2005*, Miami, Mar 2005.

[19] R. Gaeta, M. Gribaudo, D. Manini, and M. Sereno. Fluid Stochastic Petri Nets for Computing Transfer Time Distributions in P2P File Sharing Applications. *Electronic Notes in Theoretical Computer Science Copyright, 2005 Elsevier, Vol. 128, Issue 4, Pages 79-99*,

[20] D. Qiu, R. Srikant. Modeling and Performance Analysis of Bit Torrent-Like Peer-to-Peer Networks, Proceedings of ACM SIGCOMM 2004, Portland, USA, 2004.

[21] M. Adler, R. Kumar, K. Ross, D. Rubenstein, T. Suel, D. Yao. Optimal Peer Selection for P2P Downloading and Streaming, Proc. IEEE Infocom 2005, Miami FL USA, 2005.

[22] J.A. Pouwelse, P. Garbacki, D.H.G. Epema, H.J. Sips. A Measurement Study of the BitTorrent Peer-to-Peer File-Sharing System, Proc. 19th IEEE Annual Computer Communications Workshop 2004, Bonita Springs FL USA, 2004.