Content Delivery in Overlay Networks: a Stochastic Graph Processes Perspective

Damiano Carra
Dip. di Informatica e Telecomunicazioni
Università di Trento, Trento, Italy
carra@dit.unitn.it

Renato Lo Cigno
Dip. di Informatica e Telecomunicazioni
Università di Trento, Trento, Italy
locigno@dit.unitn.it

Ernst W. Biersack
Institut EURECOM
Sophia Antipolis, France
erbi@eurecom.fr

***Abstract*— We consider the problem of distributing a content of finite size to a group of users connected through an overlay network that is built by a peer-to-peer application. The goal is the fastest possible diffusion of the content until it reaches all the peers. Applications like Bit-Torrent or SplitStream are examples where the problem we study is of great interest.**

In order to represent the content diffusion process, we model the system as a stochastic graph process and define the constraints the graph evolution is subject to. The evolution of the graph is a semi-Markov process where the sojourn times are the rewards of interest for the computation of the time needed to complete the file distribution. We discuss the general properties of the constrained stochastic graphs and we show preliminary results obtained with an ad-hoc Monte-Carlo technique.

I. Introduction

The peer-to-peer (P2P) networking paradigm received a lot of attention in recent years. P2P systems construct an overlay at the application layer and do not require any modification to the existing Internet. Such an ease of deployment, which is in contrast to other technologies such as IPv6 or IP-level multicast that do require modifications inside the network, makes P2P systems very attractive for supporting new communication paradigms.

One of the most popular P2P application is file sharing, a variation of which can also be used for file distribution. P2P for file distribution has the appealing feature of self-scaling: the more are the users downloading the content the larger the overall amount of resources available for the entire system.

In this paper we consider the specific problem of how to distribute a file to a community of users organized as an overlay of peers: what is the most efficient architecture and protocol that can be used to distribute this content to the users? Applications include, for instance, distribution of virus footprints or software updates. We focus the analysis to files distributed to cooperative users willing to receive and forward it to others.

We give a formal definition of the process underlying the construction of a 'distribution graph' as a semi-Markov process, describing how different choices impact the structure of the stochastic process itself (and obviously the constructed graph) as well as the rewards used to derive the performances. Then, we analyze the properties of the semi-Markov processes. Finally, we discuss some results. The representation of content delivery overlay networks through stochastic graph processes allows giving a high level description of different kind of protocols and architectures and comparing them, abstracting from the implementation details.

A. Related Work

Performance analysis in terms of the minimum time required to distribute a file using a P2P system has only in recent years received some attention. Most of the analytical work [1][2] focuses on a specific system and not on a generic distribution architecture, and they assume strict hypotheses, like the uniformity of access bandwidths. Only [3] tackles the problem of bandwidths heterogeneity, which is also treated in this paper.

To the best of our knowledge, very few models have been proposed that allow comparative studies of different distribution architectures. In [4], inspired by SplitStream [10] (an overlay streaming protocol) the authors have defined and analyzed linear chain and tree-based architectures assuming ideal conditions, hence in a completely deterministic situation. The work in [5] defines a model for chain-based and tree-based architectures and analyzes the system using max-plus algebra considering an infinite number of packets, calculating the long term average throughput; our analysis instead considers a finite file size and calculates the distribution of the download time of all the peers.

Stochastic graph processes, the analytical tool we use in this paper to model overlay content delivery networks, were defined in [6] with the same notation we use here, although they were known since the '50s. A sub case, the *random graphs* [7], were studied in detail obtaining their general properties. The focus of the analysis in [7] is the topological properties of random graphs, whereas our aim is to take into account not only connections among nodes, but also their weights given by the bandwidths of the involved nodes, which give rise to the state reward structure that allows the computation of completion times.

II. Problem Formulation

The aim of the service is the delivery of a given finite size content $\mathcal{F}$ to a set of users. The only requirement of the service is content integrity and the main performance metric is the *download time* T of the content, either for a given user i (T_i), or for the whole community (T_t), or the mean $\overline{T}$ of all the individual download times T_i.

We assume that each node knows a subset of the whole community, i.e., a node has a finite number of neighbors.

The content $\mathcal{F}$ is divided in C pieces called *chunks*. A chunk represents a basic unit of transmission that can be distributed independently. During the distribution of $\mathcal{F}$, a node that has started to download $\mathcal{F}$ can in turn start to upload $\mathcal{F}$ after entirely receiving the first chunk.

For each node i, we define b_i^u and b_i^d as the upload and download bandwidth respectively, which can be either symmetric, asymmetric or correlated, e.g., $b_i^u + b_i^d$ constant, as in a shared medium based access. The bandwidths are random variables described by a probability density function (pdf) that is known.

When a node starts uploading chunks of $\mathcal{F}$, the *effective rate* used to transfer the chunks to each child is computed according to the *max-min fairness* criterion. The rate depends on multiple factors, such as the number of children of the uploading node, the rate the uploading node is receiving, etc. While the bandwidths (upload and download) are a given, the rates are computed during the distribution process.

We define the *eligibility time* t_i^{el} of node i as the time at which node i can start uploading chunks to other nodes, i.e., it has completely received the first chunk. If a node j is child of node i and receives at a rate r_{ij}, its eligibility time is $t_j^{\text{el}} = t_i^{\text{el}} + t_{ij}$, where $t_{ij} \triangleq \frac{\mathcal{F}}{C}\frac{1}{r_{ij}}$. As a last assumption we suppose that node i chooses its children j uniformly at random among all its neighbors, not taking into account upload and download bandwidths.

A. Formalization and general definitions

The distribution of a content within a community of users can be formalized as the propagation across a graph of nodes and edges with some (stochastically defined) characteristics. Nodes are the users and edges summarize all the characteristics of the communication network between the users.

Let $\mathcal{N}$ be the set of nodes, i.e., the vertices of the graph, and $\mathcal{A}$ the set of all the arcs that connect pairs of nodes, $\mathcal{A} \subseteq \mathcal{N} \times \mathcal{N}$. We only consider fully reachable networks with bidirectional connections. $\mathcal{B}_i$ is the set of neighbors of user i, i.e., all those nodes in $\mathcal{N}$ that are known and directly reachable from node i.

The graph $\mathcal{G}(\mathcal{N}, \mathcal{A})$ represents the overlay network created, for instance, by a P2P network. The overlay layer is the basis used to form the *distribution graph*. We define the distribution graph $\mathcal{G}^*(\mathcal{N}, \mathcal{E})$ as a directed subgraph of $\mathcal{G}(\mathcal{N}, \mathcal{A})$, with $\mathcal{E} \subseteq \mathcal{A}$. $\mathcal{G}^*$ is a directed graph, since, from the content distribution point of view, the content propagates from the source to the destinations. How to obtain $\mathcal{G}^*$ from $\mathcal{G}$ is given by the rules implemented in the specific content distribution protocol. In general, we can assume that the distribution graph $\mathcal{G}^*$ is built step by step. The building process is a stochastic process and it can be modeled as a Markov chain. Let $\mathcal{N}_n^*$ be the set of nodes that belong to the distribution graph at step n, and $\overline{\mathcal{N}_n^*}$ its complement with respect to $\mathcal{N}$. The distribution graph $\mathcal{G}_{n+1}^*$ at step $n+1$ is obtained from $\mathcal{G}_n^*$ by adding a new edge $\in \mathcal{A}$ from one node in $\mathcal{N}_n^*$ to one in $\overline{\mathcal{N}_n^*}$. The complete distribution graph $\mathcal{G}^*(\mathcal{N}, \mathcal{E})$ is obtained when $\mathcal{N}_n^* = \mathcal{N}$, and $\overline{\mathcal{N}_n^*}$ is the empty set. The dynamic behavior of the distribution graph can be modeled as a *stochastic graph process*. We recall here the general definition of stochastic graph processes [6], while in Sect.III, we specialize them for the analysis of content distribution.

***Definition** 2.1:* A stochastic graph process (SGP) on a node set $\mathcal{N}$ is a discrete time Markov chain (DTMC) whose states are graphs on $\mathcal{N}$.

Even if not stated in the definition, nodes can be connected only through edges that belong to $\mathcal{A}$. Adhering to the definition given in [6], the focus is the building process and the SGP evolution implies that the graph is built step by step by adding one node and one edge at each step.

Notice that in content distribution, the distribution network (or graph) is naturally built step by step, so using the graph $\mathcal{G}^*(\mathcal{N}, \mathcal{E})$ is very appropriate. The time between two steps depends on the sojourn time of the state. If sojourn times are exponentially distributed, then we obtain a continuous time Markov chain (CTMC), but, in general, this assumption is not true and in continuous time we have a semi-Markov chain.

***Definition** 2.2:* A constrained stochastic graph process (CSGP) on a graph $\mathcal{G}(\mathcal{N}, \mathcal{A})$ is a semi-Markov chain whose states are subgraphs on $\mathcal{G}$.

From the semi-Markov chain we can derive an embedded DTMC by sampling the process exactly at transition instants.

Eligibility times t_j^{el} influence the semi-Markov process in two different ways. In the general case of randomly varying t_{ij}, they define both the transition probabilities between states and the state sojourn times. In the particular case of deterministic t_{ij} (e.g., when the bandwidth is only determined by access links), sojourn times are deterministic, and t_j^{el} define only the state transition probabilities.

The DTMC that describes a CSGP is a transient chain with adsorbing states $\in \mathcal{G}^*(\mathcal{N}, \mathcal{E})$ that are reached when $\mathcal{N}_n^* = \mathcal{N}$.

The way we defined a CSGP implies that nodes are stable and collaborative, and that the networking infrastructure is reliable enough to allow edge stability. Clearly there is the possibility of extending the analysis to cases where nodes (or edges) can disappear during the distribution process, so that $\mathcal{G}_n^*$ is derived from $\mathcal{G}_{n-1}^*$ not only by adding an edge and a node, but also by removing one node and all the edges relative to it (that may include several generations of sons).

In the following, we specialize CSGP by adding constraints that make them suitable for modeling our problem.

III. Content-Delivery CSGP

Different distribution architectures can be defined as special cases of stochastic graph processes with additional constraints. Before introducing the 'Content-Delivery constrained stochastic graph processes,' or CD-CSGP, we give some additional definitions that will simplify the characterization of each CD-CSGP.

A. Content-Delivery Related Definitions

Each node has a constraint on maximum and minimum number of active uploads (the outdegree of the node): $k_i^{\max}$ and $k_i^{\min}$.

***Definition** 3.1 (saturated node):* A node $i \in \mathcal{N}_n^*$ is called *saturated* if

- it has $k_i^{\max}$ outgoing edges that belong to $\mathcal{G}_n^*$ or
- fully uses b_i^u to transmit chunks to neighboring nodes that belong to $\mathcal{G}_n^*$.

***Definition** 3.2 (interior subset):* The subset of nodes $\in \mathcal{N}_n^*$ that at step n are saturated is called $\mathcal{I}_n$, the interior node subset at step n.

***Definition** 3.3 (leaf subset):* The set of nodes $\in \mathcal{N}_n^*$ that are not interior nodes is called the leaf node subset $\mathcal{L}_n$ at step n, with $\mathcal{L}_n = \mathcal{N}_n^* - \mathcal{I}_n$.
We consider a single node as a root of the stochastic graph. We define a distance measure based on number of hops from the root to any node i.

***Definition** 3.4 (step distance):* The number of hops from the root to a node i following the shortest path is called *step distance* or *step depth*, $d^{(i)}$.
In a tree, $\max_i(d^{(i)})$ is the tree depth.

B. Unbalanced and Uneven Trees

The general process that leads to tree-based distribution structures must abide to the following rules.

***CD-CSGP** 1:* A constrained stochastic graph process on graph $\mathcal{G}(\mathcal{N}, \mathcal{A})$ is called tree-based if

1) $\mathcal{G}_0^*$ is a node, called root, randomly chosen in $\mathcal{N}$.
2) $\mathcal{G}_n^*$ is obtained from $\mathcal{G}_{n-1}^*$ by
   a) choosing the node i from $\mathcal{N}_{n-1}^*$ with the smallest eligibility time: $t_i^{\text{el}} = \min_j(t_j^{\text{el}})$; if more nodes have the same eligibility time, the choice among these nodes is made randomly;
   b) adding edges from node i to nodes randomly chosen from $\mathcal{B}_i \bigcap \overline{\mathcal{N}_{n-1}^*}$, until node i becomes saturated.

Figure 1 shows an example with few states of the DTMC generated by a CD-CSGP 1 process. In this case we have only two possible bandwidths (slow nodes with black circles, fast nodes with white ones, with slow bandwidth less than half of the fast bandwidth) and $k_i^{\max} = 2$ and $k_i^{\min} = 1$. Starting from a state where the server is uploading to a slow and to a fast node, the fast node has the smallest eligibility time and there are only three next possible states: (i) the fast node selects a fast nodes among its neighbors, becoming saturated; alternatively, the fast node chooses a slow node so it has to select another node: (ii) the selected node is fast and we have bandwidth saturation, or (iii) the chosen node is slow and we have saturation because $k^{\max}$ is reached. Note that in case (i) the node becomes saturated since the rate of the content it is receiving is high. If, for instance, the rate were slow (consider the fast node under the slow node in the shadowed state), the number of children would be in any case 2, since the bandwidth used to transmit to each child is at most equal to the rate it is receiving.

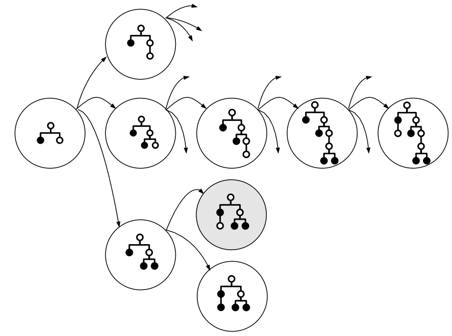

Fig. 1. Sample of the embedded DTMC for a CD-CSGP 1 process; states are graphs built on $\mathcal{G}$, black and white circles represent slow and fast nodes respectively, $k_i^{\max} = 2$ and $k_i^{\min} = 1$.

The resulting tree is, in general, a structure where the nodes in the leaf set $\mathcal{L}_n$ do not all have the same step distance from the root. The speed of growth of the different branches is not the same. And the deeper branches are those that contain faster nodes, i.e., nodes with smaller eligibility times t^{el}. We call such trees "*uneven*."

The majority of the works on tree-based distribution architectures consider trees where leaves have the same distance from the root. We call such trees "*unbalanced*." The difference between unbalanced and uneven trees is enormous: in an unbalanced tree, a slow node will influence the reception of all nodes in its subtree, in an uneven tree, a slow node may not even have the possibility to have children. Since we are interested in the download time, it is worth to look at a weighted graph where the weight associated to a directed edge is given by the difference between the download times of the nodes connected by the edge. In unbalanced trees, this representation shows the disparity in terms of download time among leaf nodes that are at the same step distance. In Fig. 2 the weight is represented as a difference in edge length. Conversely, in uneven trees, leaf nodes are at different step distances and the weighted graph gives a pictorial illustration why the tree grows in this way: a new edge is added only after a node becomes eligible and this forces a uniform growth of the *weighted* graph.

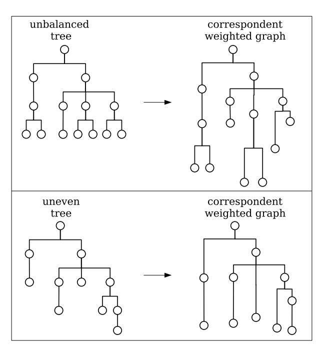

Fig. 2. Difference between unbalanced and uneven trees, considering the corresponding weighted graphs where edge length represents the download time.

We define a special stochastic graph process for this type of tree. To do so, we consider a subset of the leaf set $\mathcal{L}_n$.

Definition *3.5:* Let $d_n^{\text{MAX}} = \max_j(d_n^{(j)})$ be the maximum step distance of the nodes $j \in \mathcal{L}_n$. The subset $\widetilde{\mathcal{L}_n} \subseteq \mathcal{L}_n$ is defined as follows: $\widetilde{\mathcal{L}_n} = \{\ i \in \mathcal{L}_n \mid d_n^{(i)} < d_n^{\text{MAX}}\}$.
Now we can define the process that leads to unbalanced trees.

CD-CSGP *2:* A constrained stochastic graph process on graph $\mathcal{G}(\mathcal{N}, \mathcal{A})$ is called tree-based and unbalanced if

1) $\mathcal{G}_0^*$ is a node, called root, randomly chosen in $\mathcal{N}$.
2) $\mathcal{G}_n^*$ is obtained from $\mathcal{G}_{n-1}^*$
    a) choosing a node i from $\widetilde{\mathcal{L}_{n-1}}$ if not empty (otherwise from $\mathcal{L}_{n-1}$) with the smallest eligibility time; if more nodes have the same eligibility time, the choice among these nodes is made randomly;
    b) adding edges from node i to nodes randomly chosen from $\mathcal{B}_i \bigcap \overline{\mathcal{N}_{n-1}^*}$, until the node becomes saturated.

We will show in Sect. V the impact of unbalanced and uneven tree on the performance.

C. General Mesh Architecture

Tree based architectures have known shortcomings. Each node has only one ancestor and in case of a nodes failure, the entire subtree will stop receiving data. Each node must divide the upload bandwidth among its children, so children use only a fraction of their download bandwidths for receiving chunks; if we consider the case of asymmetric capacities, where the upload bandwidth is smaller than the download bandwidth (as in the case of ADSL), the percentage of unused download bandwidth increases even further. Finally, leaf nodes receive the entire file without uploading a single chunk, resulting in unfairness and poor performance.

Mesh based architectures are meant to overcome these problems. Nodes can upload to other nodes already reached by the content. In this case we have to consider the 'freshness' of the information that a node is downloading from its fathers.

We assume that the server, that is the only node that has the full content and can differentiate the parts to distribute, gives the chunk in different orders to its children, but different techniques such as network coding can be envisaged. We define *diffusion trees* the trees generated by these children. Nodes can receive the content from different fathers provided that these fathers belong to different diffusion trees. In general, if the server has k_s children, each node can have up to k_s fathers, each of them having $\mathcal{F}/k_s$ fresh content. For a detailed description of stochastic graph processes for mesh based networks refer to [11] for the lack of space.

IV. CD-CSGP Solution

The CD-CSGP we have defined in Sect. III can describe different behaviors of content distribution protocols and algorithms. Consider for example the two well known application layer multicast protocols, ALMI [9] and SplitStream [10], which use different topologies for the distribution. The first builds a tree structure to deliver the content, the second a mesh structure. Through CD-CSGP we can compare these two approaches, abstracting from any protocol detail.

A. Properties of CSGP and rewards

Let $\mathbf{S}$ be the state space of the DTMC embedded in the CD-CSGP we consider; $S_k \in \mathbf{S}$ are the states of the DTMC, i.e., the graphs $\mathcal{G}_n^*$. We start considering trees: to compute the mean download time $\overline{T}$ of $\mathcal{F}$ we assign to each absorbing state $S_k \in \mathbf{S}_a$ ($\mathbf{S}_a$ is the set of absorbing states) a reward $\overline{T}_k$ equal to the mean download time of the nodes in the state,

$$\overline{T}_k = \frac{1}{|\mathcal{N}_n^*|} \sum_{i \in \mathcal{N}} T_i \ ; \ S_k \in \mathbf{S}_a$$

The mean download time $\overline{T}$ is the reward of a DTMC obtained by adding a deterministic transition from all the absorbing states to an initial state represented by the empty graph $\emptyset$.

$$\overline{T} = \frac{\sum_{k \in \mathbf{S}_a} \overline{T}_k \pi_k}{\sum_{k \in \mathbf{S}_a} \pi_k} \ .$$

where π_ks are the steady state probabilities of the support DTMC defined in Sect. II-A.

Another performance measure easily defined as a reward is the wasted upload bandwidth $\overline{w^u}$ (in percentage). Let $w_i^u = 100(1 - \frac{\max(r_i^u)}{b_i^u})$ be the wasted upload bandwidth of node i, where $\max(r_i^u)$ is the maximum upload rate ever reached by the node in any visited state. Considering again the modified DTMC and letting $\mathbf{S}_a$ be the set of absorbing states in the unmodified DTMC we have

$$\overline{w_k^u} = \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} w_i^u \ ; \ S_k \in \mathbf{S}_a \quad \text{and} \quad \overline{w^u} = \frac{\sum_{k \in \mathbf{S}_a} \overline{w_k^u} \pi_k}{\sum_{k \in \mathbf{S}_a} \pi_k} \ .$$

We developed a tool for the numerical solution of all the CD-CSGP we defined in this work. The tool is based on fast Monte Carlo techniques that allow the exploration of the chain with great efficiency, computing in the meanwhile the reliability of the produced results. It implements the algorithms driving the stochastic process providing realizations of the process. It has several configuration parameters and the outputs are the rewards associated with the process. For a detailed description of the tool we refer the interested reader to [12]. Here we only give some sample results showing the feasibility and power of CD-CSGP formalization for content delivery analysis.

V. Numerical Results

As numerical example we consider a probability density function for the node bandwidth summarized in Table I.

TABLE I

Bandwidth distribution used in the examples

Bandwidth	% nodes
56 kbit/s	13%
640 kbit/s	23%
1.2 Mbit/s	64%

When reporting results, we normalize the data such that $\frac{|\mathcal{F}|}{\min(b_i)} = 1$ 'round', where $|\mathcal{F}|$ is the content size in bits and

$\min_i(b_i)$ is the minimum bandwidth of the input pdf in bits/s. We use a number of chunks C equal to 100. All results have confidence level 0.95 and confidence interval $\pm 10\%$.

We focus on the comparison between unbalanced and uneven trees; a detailed analysis of the influence of the different constraints, such as $k^{\max}$ and $k^{\min}$, can be found in [12].

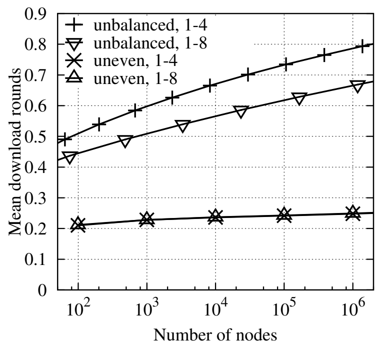

Fig. 3. $\overline{T}$ for unbalanced and uneven trees as a function of $|\mathcal{N}|$; $k^{\min} = 1$.

Figure 3 shows the results as a function of $|\mathcal{N}|$. The poorer results for unbalanced trees (CD-CSGP 2) are due to the bounds on the $d^{(i)}$. Slow nodes, especially those close to the root, impose their rate on the whole subtree. In the case of uneven trees (CD-CSGP 1), slow nodes close to the root have no time to start to upload, since the time it takes to become eligible is much more than the time it takes the fast nodes to reach, at different levels, all the other nodes. This increase of performance for the uneven tree comes at a cost of a greater step distance.

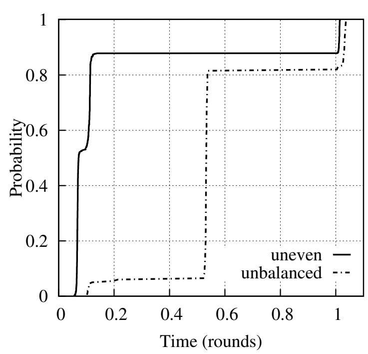

Fig. 4. CDF of T for unbalanced and uneven trees ($|\mathcal{N}| = 10^4$, outdegree 1-8).

As shown in Fig. 4, our tool provides the entire CDF (or conversely the pdf) of the download time, a result that is normally not available with analytical or semi-analytical techniques. The CDF shows clearly the reason why uneven trees are more efficient and does also show that with uneven trees fast peers are far less influenced by slow ones also increasing the perceptual fairness of users.

VI. Conclusions

In this paper we have discussed fundamental properties of content distribution systems using a novel technique based on Stochastic Graph Processes (SPG).

Describing a content distribution protocol as an SPG (with additional evolution constraints that represent the protocol-induced behavior) enables the use of powerful analysis tools based on the Monte-Carlo solution of the system equations. To the best of our knowledge, stochastic graph processes were used only to study connectivity properties, but they were never applied in performance analysis of networks, and this represent an important contribution.

Distribution systems are based on trees and meshes. Trees were studied intensively in the literature, however important details such as bandwidth heterogeneity and varying node outdegrees as well as different minimum and maximum outdegrees have received very little attention, probably for the difficulty in finding closed form results.

Our approach discloses the possibility of analyzing all the details above and many others too, and can be extended to streaming distribution and other related applications. Sample results demonstrates the power of the analytic framework we defined.

Acknowledgment

This work has been partly supported by the European Union under the E-NEXT project FP6-506869.

References

[1] X. Yang and G. de Veciana, "Service Capacity of Peer-to-Peer Networks," in *Proc. IEEE INFOCOM 2004*, Hong Kong, Mar. 2004.
[2] D. Qiu and R. Srikant, "Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks," in *Proc. ACM SIGCOMM 2004*, Portland, OR, Sept. 2004.
[3] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley, "Modeling Peer-Peer File Sharing Systems," in *Proc. IEEE INFOCOM 2003*, San Francisco, California, USA, Mar. 2003.
[4] E. W. Biersack, P. Rodriguez, and P. Felber, "Performance Analysis of Peer-to-Peer Networks for File Distribution" in *Proc. 5th International Workshop on Quality of Future Internet Services (QofIS'04)*, Barcelona, Spain, Sept. 2004.
[5] F. Baccelli, A. Chaintreau, Z. Liu, A. Riabov, S. Sahu "Scalability of Reliable Group Communication Using Overlays," in *Proc. IEEE INFOCOM 2004*, Hong Kong, Mar. 2004.
[6] S. Nikoletseas, J. Reif, P. Spirakis and M. Young, "Stochastic Graphs Have Short Memory: Fully Dynamic Connectivity in Poly-Log Expected Time," in *Proc. of the 22nd ICALP*, pp. 159-170, 1995
[7] P. Erdös and A. Renyi, "On random graphs," *Publ. Math.* 6:290–297, 1959.
[8] B. Cohen, "Incentives build robustness in BitTorrent," May 2003. Available: http://www.bittorrent.com/documentation.html
[9] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, "ALMI: An Application Level Multicast Infrastructure," in *Proc. of the 3rd Usenix Symposium on Internet Technologies & Systems (USITS)*, Mar. 2001.
[10] M. Castro, P. Druschel, A. M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: Highbandwidth Multicast in a Cooperative Environment," in *Proc. ACM Symposium on Operating Systems Principles (SOSP 03)*, The Sagamore, New York, USA, Oct. 2003.
[11] D. Carra, R. Lo Cigno, and E. W. Biersack, "Stochastic Graph Processes for Performance Evaluation of Content Delivery Applications in Overlay Networks," Technical Report DIT-06-013, Univ. of Trento, Feb. 2006. Available: http://www.dit.unitn.it/locigno/preprints/DIT-06-013.pdf
[12] D. Carra, R. Lo Cigno, and E. W. Biersack, "Fast Stochastic Exploration of P2P File Distribution Architectures," Technical Report DIT-06-014, Univ. of Trento, Feb. 2006. Available: http://www.dit.unitn.it/locigno/preprints/DIT-06-014.pdf