



# UNIVERSITÀ DEGLI STUDI DI TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

---

38050 Povo — Trento (Italy), Via Sommarive 14  
<http://dit.unitn.it/>

## ON THE FUNDAMENTAL PROPERTIES OF MESH-BASED OVERLAY STREAMING SYS- TEMS

D. Carra, R. Lo Cigno and E.W. Biersack

July 2006 – Ver. 1.0

Technical Report # DIT-06-043

# Graph Based Analysis of Mesh Overlay Streaming Systems

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**—This paper addresses the study of fundamental properties of stream-based content distributions services. We assume the presence of an overlay network (such as those built by P2P systems) with limited connectivity degree, and we develop a mathematical model that captures the essential properties of overlay-based streaming protocols and systems.

The methodology is based on graph theory and models the streaming system as a stochastic process, whose characteristics are related to the streaming protocol. The model captures the fundamental properties of the streaming system, such as the number of active connections, the different play-out delays of nodes and the probability of not receiving the stream due to nodes failures/misbehavior. Besides the static properties, the model is able to capture the transient behavior of the distribution graphs, i.e., the evolution of the structure over time, for instance in the initial phase of the distribution process.

Contributions of this paper include the detailed definition of the methodology, its comparison with other analytical approaches, and a discussion of the additional insights enabled by this methodology. Results show that mesh based architectures are able to provide bounds on the receiving delay and maintain rate fluctuations due to system dynamics very low. Additionally, given the tight relationship between the stochastic process and the properties of the distribution protocol, this methodology gives basic guidelines for the design of such protocols and systems.

## I. INTRODUCTION

The recent success of streaming based on peer-to-peer (P2P) applications seems to achieve what traditional streaming and multicasting applications have never achieved: distributed video-on-demand and live broadcasting on the Internet. The first tree based systems [1][2][3] coexist now with more advanced mesh-based systems [4][5][6] that are more resilient to node dynamic behavior and more suited to the intrinsic Internet characteristics.

In spite of the success of P2P streaming, the fundamental properties of such systems have not been investigated in depth (see Sect. I-A for a discussion of existing works). In particular, we are not aware of any study concerning the behavior of the streaming distribution system as a function of the topological properties of the graph that is built by the P2P application.

In this work we develop a mathematical model based on graph theory that can be used to analyze fundamental performance issues of overlay streaming services. We model such systems with a high level abstraction that allows the study of fundamental behavior under different conditions. Considering graph theory, many studies analyze *static* graph, identifying

properties of a snapshot of the network. In our work, instead, we study the *dynamics* of the graphs, i.e., the evolution of the structure over time.

We derive the master equations that define the evolution of the streaming system in time, based on the basic characteristics of the streaming protocol as well as the bandwidth available at nodes for the streaming application. The model allows to assess the impact of different protocol choices, and of bandwidth heterogeneity on the delivery process and gives enough insights in the problem to formulate improved streaming strategies.

A fast and effective Monte Carlo integration methodology is implemented to solve the mathematical model. The solution provided by this method is compared with other modeling and solution techniques to show the flexibility of the approach.

The results obtained by the systematic study of different configurations and scenarios show that performances are mainly influenced by the policies related to content format. Mesh based architecture are very robust to failures, even in presence of 50% of churn and the delay experimented by nodes is bounded.

The remainder of this paper is structured as follows. Sect. II introduces mesh based streaming systems. In Sect. III we recall the mathematical background used to model the system and in Sect. IV we describe in detail the analytical model. Sect. V discusses the solution approach. We present the results in Sect. VI and conclude the paper with some additional discussion in Sect. VII.

### A. Related Work

In the last few years many solutions have been proposed for overlay streaming services, also known as Application Level Multicast (ALM). Such systems can be classified according to the basic structure they use to deliver the content.

Systems such as ALMI [1], NICE [2] and Zigzag [3] organize the nodes following a tree structure. The stream is received from a single father and uploaded to a set of children (if any). The differences among these systems concern algorithms used for managing the structure in case of node dynamics. Systems such as Narada [4], Coolstreaming [5] PRIME [6] and PULSE [7] leverage on mesh structure. Nodes download from a set of fathers that can change over time. Problems related to delay and synchronization are handled

according to different heuristics. Other systems adopt an hybrid approach (e.g., SplitStream [8]), where the stream is distributed using multiple trees obtaining a structured mesh.

The above distribution protocols were not designed with a performance oriented approach as far as delivery is concerned. Many proposals use heuristic methods to improve performance, but these heuristics are verified a posteriori and protocol parameters are tuned according to these results. Performance analysis of overlay streaming systems received some attention only recently. Most of the analytical works focus on tree based structures (e.g., [9]) or on a specific system, but, to the best of author's knowledge, no study has been done on modeling general mesh-based streaming systems. Only [6] starts analyzing such systems, but with a simulative approach and assuming a homogeneous access bandwidth.

Our model considers the properties of the overlay graph. Many studies related to graph theory focus on the properties of growing networks [10][11], but the way the graph structure can grow is not constrained by protocol rules. In our approach, we specialize the general concepts of such theoretical model to our problem.

## II. MESH-BASED OVERLAY STREAMING SYSTEMS

In this section we give a high level description of a generic mesh-based overlay streaming system: we do not consider a specific system but we identify common basic characteristics of recent proposals [5][6][7]. Consider an overlay network built by a P2P application. Once the overlay layer is built, according to the rules of the distribution protocol, paths between the source and the destinations are created. At each hop, nodes both receive the stream and contribute uploading it to other nodes, i.e., they work as content relay. Since nodes in such networks can appear or disappear frequently, the set of nodes from which a node  $i$  is downloading changes over time.

The content is distributed using  $R$  different stripes. Each stripe contains part of the stream (coded, for instance, using MDC techniques [12]). A node needs  $R' < R$  out of  $R$  stripes to achieve a target quality, while the remaining  $R - R'$  stripes contain redundant information. We assume that each node downloads a specific stripe  $R_i$  from a single node, since downloading the same stripe from multiple fathers can increase the resilience to father failure, but not the quality of the received stream.

Even if the structure is a mesh, looking at the system at a specific instant  $t$ , it is possible to identify sub-structures inside the mesh. If we consider the graph at time  $t$  and we focus on the nodes that are downloading a stripe  $R_i$ , it is possible to draw a tree that connects such nodes. The whole mesh can be seen as an intricate overlap of trees, changing over time. In Fig. 1 we show an example where we take two snapshots of an overlay graph before and after a node disappears. There are two stripes and node 7 contributes to upload only one stripe (Fig. 1a). The Graph of the complete mesh is given in Fig. 1c. When node 7 leaves, the whole subtree under this

node disconnects and each node tries to find other fathers with the same stripe. The configuration of the tree has completely changed and nodes that belonged to the same subtree now are in different subtrees (e.g., node 5 and 6). We call *diffusion tree* the tree that distributes a stripe at certain instant  $t$ .

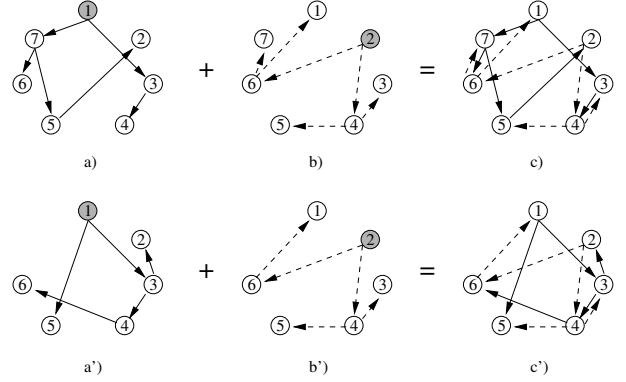


Fig. 1. Detailed view of a mesh in two instants: before (upper row) and after (lower row) node 7 leaves.

### A. System parameters

The evolution of the network is subject to two main events: node arrivals and departures. We assume that arrivals and departures are exponentially distributed according to rates  $\lambda(t)$  and  $\mu(t)$  respectively. The dependence on the time makes the model more flexible: for instance, we can describe different arrival patterns, such as flash crowds or more smooth arrivals. Let  $T_{\text{str}}$  the duration of the stream and  $N$  the mean number of nodes at the end of the stream. The arrival pattern is composed by the initial percentage of nodes present in the system when the stream starts and the interval within the remaining nodes arrive. Fig. 2 shows an example of the target mean number of nodes in the system, from which parameters ( $N$ , initial nodes, arrival interval) it is possible to derive  $\lambda(t)$ .

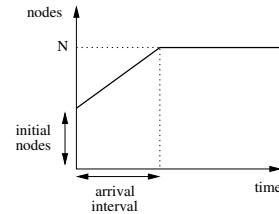


Fig. 2. Arrival pattern.

The departure rate  $\mu(t)$  represents the inverse of the mean time spent in the system (sojourn time). Since we want to maintain the mean number of nodes according to the arrival pattern as in Fig. 2, the arrival rate is adjusted with an additional term that compensates the departure rate. The ratio between the cumulative number of left nodes and the cumulative number of arrived nodes is defined as the *churn* of the system. We can study extreme cases where the sojourn time is smaller than the stream length, i.e., very high churn.

Nodes are divided into different classes according to their bandwidth. Each class  $j$  has an upload bandwidth  $b_u^{(j)}$  and

a download bandwidth  $b_d^{(j)}$ , 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 (e.g., derived from measurement studies).

The rate of the streaming is  $r_{\text{str}}$ . We suppose that all nodes have a download bandwidth at least equal to the streaming rate. Each stripe has a rate equal to  $r_{\text{str}}/R'$ , and we assume that the server is able to upload all the  $R$  stripes, i.e., it has a bandwidth greater than  $Rr_{\text{str}}/R'$ . Each node has a constraint on maximum and minimum number of active uploads that limit the possible outdegree of the node:  $k^{\text{max}}$  is the *maximum outdegree* and  $k^{\text{min}}$  is the *minimum outdegree*. As regard the indegree, a node can download at most  $R$  different stripes.

Each node has  $B$  neighbors. Among its neighbors the node will select the nodes from which download (fathers). The first  $R'$  fathers are called *active* fathers; the remaining fathers are called *standby*, since they are used as a backup in case of active father failure<sup>1</sup>.

### B. Join, Update and Leave Procedures

Nodes belonging to the initial set start building a diffusion tree for each stripe. The number of nodes in each diffusion tree depends on the characteristics of the nodes involved, i.e., their access bandwidths that determine the possible children. Each node is involved in multiple diffusion trees.

When a new node arrives, it chooses randomly an active node as first contact, and then builds its neighbor list iteratively: it takes a random neighbor of the contact node, then a random neighbor of the random neighbor of the contact node, and so on, until it adds  $B$  neighbors. From its new neighbors list, it selects its fathers and attaches to them: the selection process is done following the constraints on maximum number of indegree/outdegree, bandwidth saturation and received stripes.

With rate  $\lambda_{\text{update}}$  (exponentially distributed) nodes periodically search among their neighbors new connections in order to increase their indegree. For standby fathers the bandwidth is not reserved, so the total number of father can exceed the ratio between the stripe rate and the node download bandwidth.

When a node leaves, all the inbound and outbound connections are canceled. Orphan nodes try to replace the left father. If the left father was in the standby set, the node does not react (it simply loses a backup father). If the left father was in the active set, the node tries to switch the state of a standby father, i.e., it starts downloading from the standby father if it has available bandwidth. If the node has no backup fathers, there are different *reaction policies*:

- *Fast Reaction (FA)*: the node searches immediately among its neighbors for new connections, i.e., it starts an update procedure. This will result in a temporary loss of quality that depends on the time necessary to search for a new father;

- *No Reaction (NR)*: the node waits until the next update procedure (scheduled, on average, every  $1/\lambda_{\text{update}}$  seconds). The interval during which the node receives poorer quality depends on  $\lambda_{\text{update}}$ .

The aim of NR policy is to limit overhead messages sent over the networks.

## III. MATHEMATICAL BACKGROUND

The network of contacts among users of a P2P networks can be modeled as a graph, where nodes represent the users and edges the neighborhood relationship. When the users start exchanging data (in our case, they start receiving and distributing the stream) they use a subset of the available outgoing/incoming edges. The number of used incoming neighbors, for instance, represents the number of fathers from which they download the content, and can be considered as a metric to measure the total rate received and consequently the quality of the streaming. The focus of our analysis is the characteristics of the *distribution graph* (see Fig.3), i.e., the subgraph of the overlay graph, where edges are the connections effectively used by nodes.

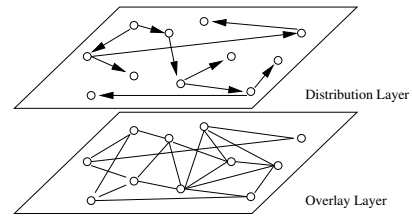


Fig. 3. Overlay and distribution graphs

Let  $\mathcal{G}(\mathcal{N}, \mathcal{E})$  the distribution graph, where  $\mathcal{N}$  is the set of nodes (users) and  $\mathcal{E}$  the set of (directed) edges that connect pairs of nodes. In general,  $\mathcal{G}(\mathcal{N}, \mathcal{E})$  is time varying, i.e., nodes and edges can appear or disappear in time. The evolution of the graph can be seen as a stochastic process with Markovian properties, since the graph at time  $t + dt$  depends only on the graph at time  $t$  and the event (join or leave of a node) occurred during  $dt$ .

### A. Distribution Graph Properties

The distribution graphs can be described through their structural characteristics. Given the arrival and departure processes, and considering the building rules the graph is subject to, we can derive, besides the mean number of nodes participating in the streaming, two main distributions: the degree distribution and the delay distribution [10].

The degree distribution  $p_s(k, t)$  is the probability that node  $s$  has  $k$  connections at time  $t$ . Since the distribution graph is directed, we are interested in the indegree and outdegree distributions ( $p_s(k_i, t)$  and  $p_s(k_o, t)$  respectively, with  $k_i + k_o = k$ ), that represent the number of children and the number of fathers of node  $s$ . We refer in general to the degree distribution implicitly assuming that we are interested in the indegree and outdegree distributions. Knowing the degree distribution

<sup>1</sup>We borrowed the concept of grouping active and standby fathers from [13].

of each node in the graph, we can derive the total degree distribution

$$P(k, t) = \frac{1}{N(t)} \sum_{s=1}^{N(t)} p_s(k, t) \quad (1)$$

where  $N(t)$  is the number of nodes attached to the streaming at time  $t$ .

The delay distribution represents the distance of the node from the source of the stream following the shortest path. We define  $p_s(l, t)$  as the probability that node  $s$  is  $l$  steps away from the source at time  $t$ . Similarly to the degree distribution, we can derive the total delay distribution

$$P(l, t) = \frac{1}{N(t)} \sum_{s=1}^{N(t)} p_s(l, t). \quad (2)$$

Hereinafter, for notation simplicity, given a value  $\alpha$  for the degree or the delay, we will use the lower case for the probability of a single node  $p_s(\alpha, t) = p(\alpha, t)$ , omitting the specification of node  $s$ , whereas we will use the upper case  $P(\alpha, t)$  for the probability of the total number of nodes

### B. Master Equations and Rate Equations

The analysis of the evolution of the graph can be done through the study of the evolution of the properties defined in the previous section, the degree distribution and the delay distribution. For a Markov process, the temporal behavior can be described using the differential form of the Chapman-Kolmogorov equations, known as *Master Equations* (MEs) [10].

Considering a node  $s$ , the variation of the probability to find the value  $\alpha$  ( $\alpha$  represents the degree or the delay) at time  $t$  can be expressed as

$$\frac{\partial}{\partial t} p(\alpha, t) = \sum_{\beta} w_{\beta, \alpha}(t) p(\beta, t) \quad (3)$$

where  $w_{\beta, \alpha}(t)$  represents the transition rates from the value  $\beta$  to the value  $\alpha$  at time  $t$ . The transition rates are closely related to the streaming protocol policies and behavior. The general formulation of the Master Equations must be specialized for our problem, i.e., we have to define all the possible transitions (see Sect. IV).

The MEs fully determine the evolution in time of the stochastic system for any node  $s$ . It is also useful to have the equations for the average value (degree  $\bar{k}$  or delay  $\bar{l}$ ). The correspondent equations are called *Rate Equations* (REs):

$$\frac{\partial}{\partial t} \bar{\alpha} = \frac{\partial}{\partial t} \sum_{\alpha} \alpha p(\alpha, t) \quad (4)$$

The Rate Equations describe the average quantities and express deterministically the behavior of the system: actually, REs are a set of differential equations that describe the evolution over time of the mean properties of the system. Figure 4 shows the relationship between the results of the MEs and the result of the REs for a given observed random variable (e.g, node degree or delay). MEs clearly provide a great insight on the system (at

TABLE I  
NOTATION AND MODEL PARAMETERS.

|                           |   |
|---------------------------|---|
| $\bar{\alpha}$            | mean value of quantity $\alpha$   |
| $k$                       | node degree   |
| $k_i$                     | indegree  |
| $k_o$                     | outdegree   |
| $l$                       | delay (or number of steps) w.r.t. the source                            |
| $N(t)$                    | number of nodes in the network  |
| $\lambda$                 | arrival rate  |
| $\mu$                     | leave rate  |
| $\lambda_{\text{update}}$ | rate when node updates indegree   |
| $p_a$                     | probability to attach to a diffusion tree                               |
| $p_a(m)$                  | probability to attach to $m$ diffusion trees                            |
| $p_c(m)$                  | probability to become a child of a node while searching for $m$ fathers |
| $R$                       | number of stripes   |
| $R'$                      | number of stripes necessary for basic quality                           |
| $R^{(j)}$                 | maximum number of stripes that class $j$ can receive                    |
| $\bar{k}_{\text{diff}}$   | mean outdegree in the diffusion trees                                   |
| $\bar{h}_{\text{diff}}$   | mean depth of diffusion trees   |

a cost of more resources necessary to find the solution), since they fully characterize the properties over time. REs give a mean value that is equivalent to fluid approximation of the system. As the time goes to infinity, MEs converge to the steady state distribution, whereas REs converge to the mean value.

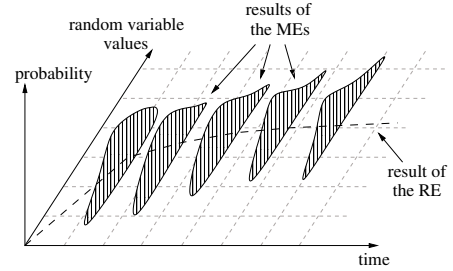


Fig. 4. Results of the Master Equations and the Rate Equations

The methodology we propose is able to provide the solution for the MEs, and hence the complete system characterization. When the complexity of the system increases and the required resources for solving it become prohibitive, we can focus on the REs and obtain an analysis of the mean value. Thus, the proposed solution method offers a great flexibility in deciding the desired level of detail in the system analysis.

## IV. DETAILED MODEL DESCRIPTION

In this section we describe the transition rates for the different graph properties, degree and delay. The MEs and the correspondent transition rates can be written for all the classes of nodes. For notation simplicity, we do not state explicitly the dependence on class  $j$  and on time  $t$ . Table I summarizes of the notation used.

### A. Indegree Distribution

Consider the basic MEs for the indegree distribution  $k_i$  for a generic class  $j$ . The transition rates in case of NR policy

$w_{k'_i, k_i}^{\text{NR}}$  can be expressed as follows:

$$w_{k'_i, k_i}^{\text{NR}} = \begin{cases} 0 & \text{if } k'_i > R \text{ or } k_i > R \\ k'_i \mu & \text{if } k'_i = k_i - 1 \\ p_a(k_i) & \text{if } k'_i = 0 \\ \lambda_{\text{update}} p_a(k_i - k'_i) & \text{if } 0 < k'_i < k_i \end{cases} \quad (5)$$

The first term states that each node has at most  $R$  fathers, so no transitions are possible from  $k'_i > R$  or to  $k_i > R$ . The second term considers the case in which one of the  $k'_i$  fathers may leave (each with individual rate  $\mu$ ). The last two terms represent the cases when the node joins and when it periodically tries to update its indegree. Let  $p_a(m)$  be the probability to attach to  $m$  diffusion trees (during the join or update procedures the node can find multiple fathers at a time). When the node joins, the indegree is given by  $p_a(k_i)$ ; when it performs the indegree update, with rate  $\lambda_{\text{update}}$ , the indegree is given by  $p_a(k_i - k'_i)$  since the node has already  $k'_i$  fathers.

In order to compute  $p_a(m)$ , we have to find  $p_a$ , the probability to attach to a single diffusion tree. We assume that all nodes receive  $R'$  stripes, so the number of active nodes in each tree is  $\frac{R'}{R}N$ . Each node has a number  $B$  of neighbors and the probability that the  $\frac{R'}{R}N$  nodes of a tree are in the node's neighbor set is:

$$p_a = 1 - \left(1 - \frac{\frac{R'}{R}N}{N}\right)^B = 1 - \left(1 - \frac{R'}{R}\right)^B. \quad (6)$$

In fact, the probability to select a node of the tree is  $\frac{R'}{R}$ , and the probability to fail to select such node for  $B$  times is  $\left(1 - \frac{R'}{R}\right)^B$ .

The probability to attach to a diffusion tree is independent for all the trees, thus the probability to attach to  $m$  trees at the same time is given by the Binomial distribution with parameters  $R - m'$  (where  $m'$  is the number of fathers that the node already has) and  $p_a$ , i.e.,

$$p_a(m) = \binom{R - m'}{m} p_a^m (1 - p_a)^{(R - m') - m}$$

The transitions rates in case of FA policy,  $w_{k'_i, k_i}^{\text{FA}}$ , are similar to Eqs. (5) excepting  $\lambda'_{\text{update}}$ , instead of  $\lambda_{\text{update}}$ , defined as

$$\lambda'_{\text{update}} = \begin{cases} \lambda_{\text{update}} & \text{if } k_i > R' \\ \lambda_{\text{update}} k'_i \mu & \text{if } k_i < R' \end{cases} \quad (7)$$

In fact, a new update event is performed after each active father departure.

### B. Outdegree Distribution

We first compute the number of nodes (active and standby) in each tree. The total number of edges in the overlay graph is given by the mean number of indegree  $\bar{k}_i$  multiplied by the number of nodes  $N$ . Note that the mean indegree is the global mean over all the classes. Assuming the size of the diffusion trees approximately equal, the number of nodes in each tree<sup>2</sup> is  $\frac{\bar{k}_i N(t)}{R}$ .

<sup>2</sup>In a tree, the number of nodes is equal to the number of edge plus one, and for large  $N$  we may not consider this additional node.

The transition rates of the MEs for the outdegree distribution  $k_o$  (number of children) for a generic node  $n$  belonging to class  $j$  with policy NR can be expressed as

$$w_{k'_o, k_o}^{\text{NR}} = \begin{cases} 0 & \text{if } k'_o > k_o^{\text{max}} \text{ or } \\ & k_o > k_o^{\text{max}} \\ k'_o \mu & \text{if } k'_o = k_o - 1 \\ \lambda p_c(R) & \text{if } k'_o = k_o + 1 \\ & \text{and } k'_o < k_o^{\text{max}} \\ \lambda_{\text{update}} \left(N - \frac{\bar{k}_i N}{R}\right) p_c(R - \bar{k}_i) & \text{if } k'_o = k_o + 1 \\ & \text{and } k'_o < k_o^{\text{max}} \end{cases} \quad (8)$$

The first term states that node  $n$  has at most  $k_o^{\text{max}}$  children. The second term considers that one of the  $k'_o$  children can leave (each with individual leave rate  $\mu$ ). The last two terms represent the cases when a new node joins the system, with rate  $\lambda$ , or during an update procedure: the update rate is  $\lambda_{\text{update}}$  for each node in the network, but only nodes that do not have the stripe of node  $n$  are taken into account, i.e.,  $N - \frac{\bar{k}_i N}{R}$  nodes. These rates are multiplied by the probability to become a children of a node while searching for  $m$  fathers,  $p_c(m)$ , given by

$$p_c(m) = \sum_{j=0}^m p_a(m - j) \frac{1}{m} \frac{R}{R' N} \quad (9)$$

When a node looks for a father, it has probability  $p_a(r)$  to attach to  $r$  diffusion trees. The probability that the node attach to the diffusion tree of node  $n$  is  $1/r$ . Moreover, the probability that the node chooses node  $n$  among the  $\frac{R'}{R}N$  active nodes is  $1/(\frac{R'}{R}N)$ . In case of node join, the node can select up to  $R$  fathers. In case of node update, considering that nodes have a mean indegree  $\bar{k}_i$ , they will look for  $R - \bar{k}_i$  new fathers.

The transitions for FA policy,  $w_{k'_o, k_o}^{\text{FA}}$ , are given by Eqs. (8) using  $\lambda'_{\text{update}}$ , defined in (7), instead of  $\lambda_{\text{update}}$ .

### C. Delay Distribution

The delay is represented by the distance from the source. The basic idea is to evaluate the diffusion tree depth and average outdegree, so that it is possible to compute the probability to be at a given level  $l$  of the diffusion tree. We assume for simplicity that trees are regular with mean outdegree  $\bar{k}_{\text{diff}}$ . The number of internal nodes with respect to leaf nodes in a regular tree is approximately equal to the inverse of the outdegree, i.e.,  $1/\bar{k}_{\text{diff}}$ . In a diffusion tree, each node has  $R'$  active fathers and  $\bar{k}_o$  children, so the mean number of children per tree is  $\bar{k}_o/R'$ . Only  $1/\bar{k}_{\text{diff}}$  of these children will be internal children, so from the identity

$$\bar{k}_{\text{diff}} = \frac{\bar{k}_o}{R'} \frac{1}{\bar{k}_{\text{diff}}}$$

we are able to find  $\bar{k}_{\text{diff}}$ .

The number of active nodes in a tree is  $\frac{R'}{R}N$ . The mean number of levels in each diffusion tree,  $\bar{h}_{\text{diff}}$ , can be computed solving the following identity:

$$\sum_{i=0}^{\bar{h}_{\text{diff}}} \bar{k}_{\text{diff}}^i = \frac{R'}{R} N$$

that leads to

$$\bar{n}_{\text{diff}} = \log_{\bar{k}_{\text{diff}}} \left( 1 - \frac{R'}{R}N + \frac{R'}{R}N\bar{k}_{\text{diff}} \right) - 1$$

The probability that a node is in level  $l$  is given by the ratio between the number of nodes in level  $l$  and the total number of nodes, i.e.,  $p_{\text{lev}}(l) = \frac{\bar{k}_l^{\text{diff}}}{R'N/R}$ . From this probability mass function we can derive the correspondent cumulative distribution function (CDF),  $F_{\text{lev}}$ . Each node chooses independently the position into a diffusion tree. The total delay is given by the maximum delay among all trees, so the CDF of the delay is the product of the CDFs of a single tree, obtaining

$$F_{\text{lev}}^* = \prod F_{\text{lev}} = F_{\text{lev}}^{\bar{k}_i}$$

where the last passage is done considering the independence of the  $\bar{k}_i$  probabilities described by  $F_{\text{lev}}$ .

Now we have all the elements to compute the transition rates. The transition rates of the MEs can be written as

$$w_{l',l}(t) = \mu F_{\text{lev}}^*(l) \quad (10)$$

The father that determines the delay may leave with rate  $\mu$ . The new delay is computed as the maximum among the remaining fathers.

## V. MONTE CARLO INTEGRATION OF THE MASTER EQUATIONS

The set of MEs derived in the previous section cannot in general be solved in closed form. However, the structure of the transition matrix that describes the stochastic process, is extremely suited for an efficient numerical solution based on Monte Carlo techniques [14][15][16]<sup>3</sup>, i.e., for a solution based on process simulation.

Monte Carlo integration is basically a random walk in the state space of the process. The convenience of the methodology is given by the fact that it is very simple to build a random walk following the graph building rules given in Sect. II and the same rules define a transition matrix with good local properties, i.e., given a state there are few states where the process can evolve and, from the reward point of view, they are similar one another, so that there are not “diverging paths” that may lead to instabilities in the solution.

Samples obtained via Monte Carlo techniques are i.i.d. by construction, so that confidence intervals and levels can be estimated on the whole probability distribution with high efficiency.

The key strength of the methodology we are proposing is not only the efficiency of the numerical solution. Indeed, this method provides great flexibility in the system description and specification. The realization of the stochastic process can be as close as desired to a real implementation to the protocol/system (just like in a generic simulation approach, but, being based on formal definitions, avoids the risk of incomplete or bugged specifications). For instance, many basic

<sup>3</sup>In physical and chemical sciences this technique is often called *Stochastic Simulation Algorithm* or *Gillespie Algorithm*, but we prefer to stick to the term ‘Monte Carlo’ normally used in computer science.

assumption usually made in fluid models, can be avoided using our approach, since we can describe in detail the system behavior.

The solution of MEs with Monte Carlo integration has been applied in many research fields. For some example see [17].

### A. Comparison with Other Methodologies

We consider a very simple case in order to show the differences with other modeling approaches. Consider the case of NR policy, i.e., a node update its indegree only during the update events (not if an active node fails). We assume a single class with an infinite upload and download bandwidth and no constraints on the maximum outdegree. If  $k_i(t)$  is the indegree at time  $t$ , at every update event the node will add  $R - k_i(t)$  fathers. In fact, under these assumptions the probability to find all the necessary fathers to obtain all the stripes is 1, since there is always a node that is able to provide a connection. The differential equation that describes the evolution can be written as

$$\frac{d}{dt}k_i(t) = \lambda_{\text{update}}(R - k_i(t)) - k_i(t)\mu \quad (11)$$

The second term considers the fact that the  $k_i(t)$  fathers can leave with rate  $\mu$  each. Considering the initial condition  $k_i(0) = 1$  (we suppose that all nodes are present at the beginning with exactly one father each) the solution of (11) is

$$k_i(t) = \frac{\lambda_{\text{update}}R}{\lambda_{\text{update}} + \mu} \left( 1 - e^{-(\lambda_{\text{update}} + \mu)t} \right) + e^{-(\lambda_{\text{update}} + \mu)t} \quad (12)$$

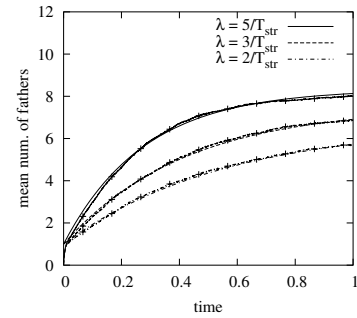


Fig. 5. Solution of the differential equation and the Rate Equation.

In Fig.5 we compare the analytical solution of this very simple case with the solution of the Rate Equations (4) derived from our model. We set  $R = 10$  stripes,  $\mu = 1/T_{\text{str}}$  and  $\lambda_{\text{update}} = \frac{5}{T_{\text{str}}}, \frac{3}{T_{\text{str}}}$  and  $\frac{2}{T_{\text{str}}}$ . We normalize the time with respect to  $T_{\text{str}}$ . The numerical solution follows closely the analytical one. But the results obtained by the numerical solution are broader. In fact we can observe how the full indegree distribution changes over time, not only its average.

Fig 6 shows the distribution of the number of fathers (indegree) at time  $T_{\text{str}}/2$  for different values of  $\lambda$ . Notice that there is a non-null probability that nodes remain with no father, thus being disconnected entirely from the distribution process, a phenomenon that a fluid approach analyzing the means entirely disregards, while in most cases it can be the most interesting

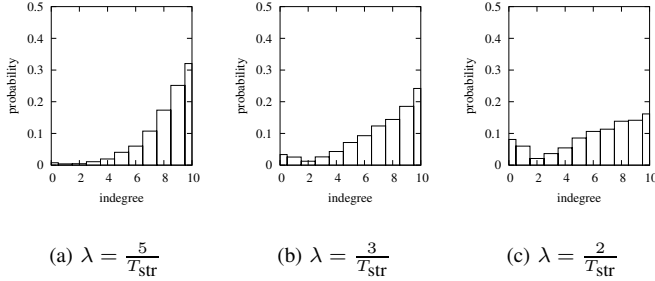


Fig. 6. Indegree distribution at time  $T_{\text{str}}/2$  obtained from the solution of the MEs.

result. This simple example shows the potential results that our approach can obtain.

## VI. APPLICATION OF THE METHODOLOGY

In this section we apply the model to study the performance of an overlay streaming system such as CoolStreaming [5] or PULSE [7]. The model we propose is not a detailed description of all the functionalities of such protocols, but a high level view of the basic mechanisms. We focus on the policies defined in Sect. II-B, Fast Reaction (FA) and No Reaction (NR), that determine the behavior in the case in which an active father leaves (node performs immediately an update procedure, or waits until the next schedule update procedure, respectively).

### A. System Description

We use a configuration with  $N = 10^4$  nodes, but we have also checked some configurations with  $10^5$  nodes obtaining similar results. We use the input bandwidth distribution reported in Table II; bandwidths are expressed as a multiple of the streaming rate  $r_{\text{str}}$ . The streaming rate is divided into  $R'$  stripes and the source generates  $R$  stripes. Results are obtained for  $R = 10$  and  $R' = 2, 4, 6, 8$ .

TABLE II  
BANDWIDTH DISTRIBUTIONS (NORMALIZED W.R.T.  $r_{\text{STR}}$ )

| Bandwidth | % nodes |
|-----------|---------|
| 1         | 20%     |
| 2         | 40%     |
| 5         | 40%     |

We consider an observation time equal to  $T_{\text{str}}$  (stream length). We consider two arrival patterns, with initial number of nodes equal to 10% and 50% of nodes  $N$ ; the remaining nodes arrive within  $T_{\text{str}}/5$ . The mean sojourn time is set to  $T_{\text{str}}$ ,  $2T_{\text{str}}$ , and  $5T_{\text{str}}$ . As regards connectivity constraints, each node can have up to 60 neighbors in the overlay graph (the actual number of neighbors depends on dynamics of the nodes); among these relationships, while uploading a node can have a maximum outdegree equal to 14. The maximum number of downloading connections is given by the number of stripes  $R$ .

The stream is chunk based (e.g., few video frames or a slice of a few tens of milliseconds of sound) and we normalize the dimension of the chunk,  $U$ , such that  $\frac{U}{r_{\text{str}}} = 1$  unit. A node becomes eligible for uploading the content after a delay equal

to the download time of a single chunk. So the delay can be considered as the “distance” (relative delay) of the node from the source of the stream. The length of the stream,  $T_{\text{str}}$ , is set to  $1000 \frac{U}{r_{\text{str}}} = 1000$  units.

Besides degree and delay properties, we consider also the *quality of the mesh*: when a node remains orphan of an active father, it switches to one of its standby father: if they have enough bandwidth to help the node, the node has no service disruption; if no standby father is able to help the node, it must search for a new father, with a possible service disruption; we measure the quality of the mesh as the percentage of nodes that successfully switch to standby father.

Space forbids to present a full set of results, with different input bandwidth distribution and constraints. We report only some representative results that show what it is possible to obtain from the analytical framework.

### B. Analysis of the Indegree

Through the analysis of the indegree we are able to examine whether the subdivision in stripes helps the distribution process or not. On the one hand, more stripes means that each stripe has a lower rate, so the loss of a single stripe has less impact. On the other hand, each node must maintain more active connections, and the probability that one of these connections fails increases.

Figure 7 shows the indegree distribution of the nodes at time  $t = T_{\text{str}}$ , computed with Eq. (1). In this case we have initial number of nodes equal to 10%, FA policy and mean sojourn time  $T_{\text{str}}$ . The distribution tends to peak around  $R$ , the number of stripes, as  $R'$  tends to  $R$ . This means that all the nodes in the network are able to receive the full quality, since the degree is always greater or equal to  $R'$ .

The temporal behavior of the indegree can be analyzed looking at the results of the rate equations (Fig. 8(a)) computed with Eq. (4). A stable value is reached after few time units: this means that the structure, even in presence of high churn is able to maintain a high quality of the stream.

The impact of the different policies (FA or NR) can be assessed looking at Figs. 8(a) and 8(b). A frequent update is necessary only if the number of backup stripes is small (e.g.,  $R - R' = 2$ ). If we have sufficient backup stripes, the performance are independent from the policy used, so updates can be made less frequently, with less overhead messages sent over the network.

### C. Analysis of the Delay

In Fig. 10 we analyze the delay in different scenarios. The delay is expressed as time units and represents the number of hops from the source. We plot the Complementary Cumulative Distribution Function (CCDF, defined as  $1 - \text{CDF}$ ) in order to study the tail of the distributions. We consider  $R' = 4$  and we set different sojourn times ( $\mu$ ). Fig. 10(a) shows the case of initial number of nodes equal to 10%, while Fig. 10(b) considers an initial number of nodes equal to 50%. In both figures, the tails of the distributions are not affected by the different values of  $\mu$ . On the other hand, comparing the two

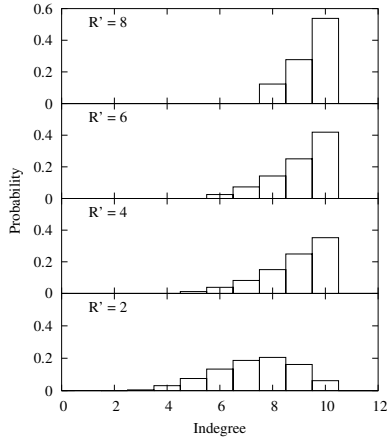
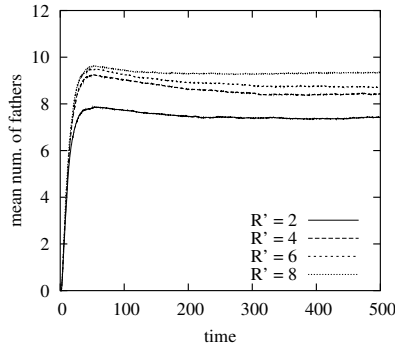
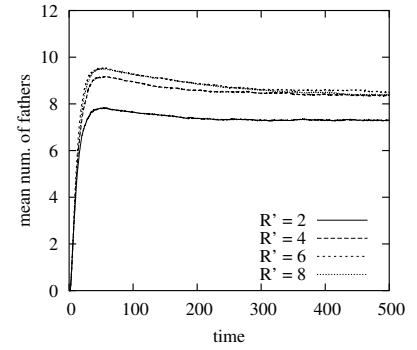


Fig. 7. Distribution of the indegree at time  $T_{str}$ .



(a) Policy FA



(b) Policy NR

Fig. 8. Indegree: evolution in time with different policies

figures, the delay are strongly influenced by the initial number of nodes.

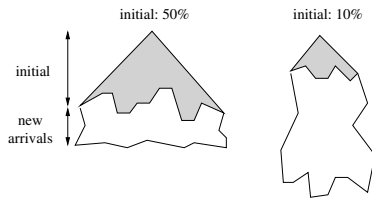


Fig. 9. Examples of diffusion trees growth with different initial number of nodes (represented with shadowed regions).

At the beginning of the distribution process, each diffusion tree grows almost regularly. As the time goes by, new arrivals attach to the leaf nodes of this structure. If the initial set of nodes is large, new nodes are able to attach to the leaf nodes with a limited increasing number of hops (see Fig. 9). If the initial set of nodes is small, new nodes form a sort of chains from the leaf nodes, with a correspondent increasing delay.

In Fig. 10(c) we show the impact of  $R'$  on the delay. Increasing the number of stripes has a side effect: since each node need all the  $R'$  stripes to correctly play the stream, the absolute delay is given by maximum delay among the stripes. By increasing the number of stripes, the probability to have higher delays increases, since we have to compute the maximum among an increased number of stripes.

#### D. Analysis of the Quality

Aggregate results for the indegree and the delay are not able to capture all the aspects related to the quality of the received stream by a generic node  $i$ . In Table III we summarize other results that can be obtained from the solution of the MEs. The value of the *churn* is computed according to the arrival pattern: arrivals and departures are Poisson processes with rate  $\lambda(t)$  and  $\mu(t)$  respectively, so we can calculate the cumulative arrived and left nodes at time  $T_{str}$  and consequently the value of churn.

During node's lifetime, there is a non null probability that all its fathers leave and node  $i$  is not able to find other fathers. This

TABLE III  
OTHER STATISTICS

| Policy | $R'$ | $1/\mu$    | % Churn  | % Error | %Switch  |
|--------|------|------------|----------|---------|----------|
| NR     | 4    | $T_{str}$  | 48.2834% | 1.0306% | 96.9571% |
| NR     | 4    | $2T_{str}$ | 31.4564% | 0.5069% | 99.0075% |
| NR     | 4    | $5T_{str}$ | 15.402%  | 0.2039% | 99.586%  |
| FA     | 4    | $T_{str}$  | 48.2227% | 1.0115% | 97.5472% |
| FA     | 4    | $2T_{str}$ | 31.4396% | 0.5112% | 99.0647% |
| FA     | 4    | $5T_{str}$ | 15.3912% | 0.2036% | 99.5929% |
| NR     | 8    | $T_{str}$  | 48.1236% | 0.9918% | 67.6836% |
| NR     | 8    | $2T_{str}$ | 31.4035% | 0.5018% | 80.4326% |
| NR     | 8    | $5T_{str}$ | 15.4263% | 0.2048% | 91.7294% |
| FA     | 8    | $T_{str}$  | 48.2136% | 0.9881% | 80.4043% |
| FA     | 8    | $2T_{str}$ | 31.3918% | 0.4862% | 85.7291% |
| FA     | 8    | $5T_{str}$ | 15.3799% | 0.1974% | 92.8333% |

situation causes an error and node  $i$  leaves the system. From the probability to have outdegree 0 at any instant  $t$ ,  $p(0, t)$ , integrating over time  $t$  we can compute the probability that a node leaves with an error message. This result is reported in column *% Error* of Table III for different policies and values of  $R'$ . With high churn, the node must disconnect from the system with probability 1%. In some contexts, this value may be considered unacceptable. For instance, if we consider a sport event and we suppose that users may disconnect temporarily during commercials (advertising), the probability of churn may reach 50%: in that situation, 1% of remaining nodes stop receiving data; users of traditional TV may find this probability too high.

Another interesting results is the probability to switch to a standby fathers if an active father leaves. This is given by  $p(k_i, t)$  with  $k_i < R'$ . Integrating over time  $t$  we are able to compute the switch probability (see last column of Table III). With a small  $R'$ , the percentage of switches is very high, i.e., the received stream is stable. On the other hand, with  $R'$  near to  $R$ , with high churn, if the number of fathers of a node  $n$  drops below  $R'$ , the probability to switch to a standby father is 67%. This means that the quality temporaliy decreases in many cases. This phenomenon is alleviated by FA policy.

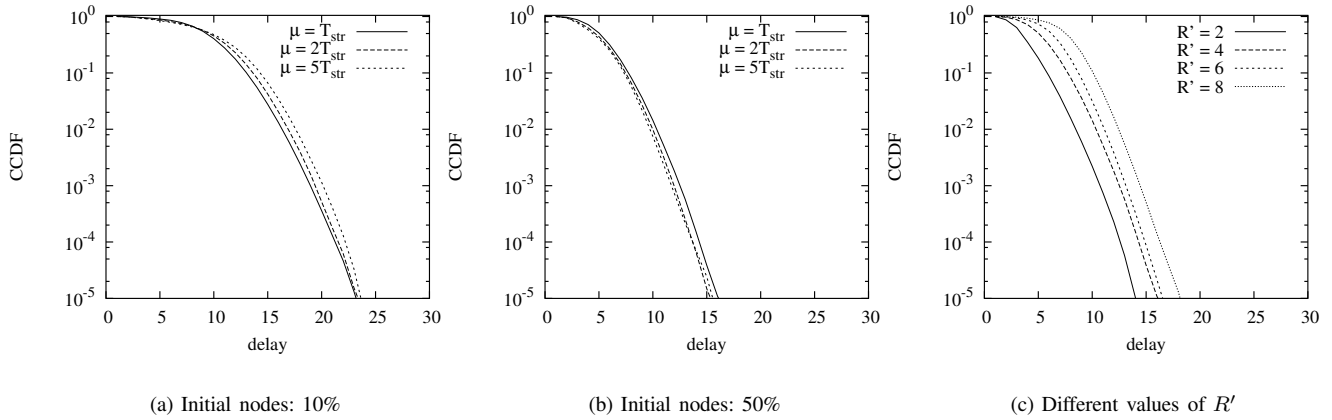


Fig. 10. CCDF of the delay in different scenarios: (a)(b) impact of initial number of nodes ( $R' = 4$ ) and (c) impact of stripe subdivision ((initial 50%, sojourn  $1/T_{str}$ ).

## VII. DISCUSSION AND CONCLUSIONS

The contribution of this paper is the introduction of a novel methodology for the high-level representation of overlay streaming systems. Based on the use of Master Equations, the solution of the model yields the entire probability distribution of the results (not only the mean value), as well as the temporal (transient) dynamics.

We have modeled some systems proposed recently obtaining novel insights in the dynamics of self-organizing systems for streaming distribution. In the following we summarize the main findings that can help in designing better P2P streaming systems.

- Redundant stripes play a fundamental role in obtaining good performances. Recent proposals [5][7] consider only a small fraction of redundant information so, in case of node departures, the stability of the streaming is affected. Maintaining standby fathers, a common solution used by such systems, may not alleviate the problem, since a node can have a standby set of fathers with the same stripes of the (remaining) active fathers. Distributing redundant stripes does not represent an overhead for the network, since nodes upload actively only the minimum amount  $R'$  of stripes necessary to reconstruct the stream.
- The delay is influenced by stripe ‘size’: the greater  $R'$  (smaller stripes) the higher the delay. The number of necessary stripes  $R'$  should be kept low to keep a low delay. The delay remains low independently from the dynamics of the network, which is a counter-intuitive result.
- If the system cannot support redundant stripes, it has to implement policies that limit the variability of the number of fathers, as we have shown with FA policy. This comes at a cost of increased overhead messages in the network.
- Under medium to high churn, nodes may become disconnected from the stream, interrupting the service (notice that instead churn does not affect delay). Only stable nodes can prevent disconnections. This performance measure cannot be computed with any methodology that only yields averages, and may be difficult to observe

with standard simulations, because even small (say  $< 1\%$ ) disconnection rates are unacceptable and require simulating thousands of nodes for hours to be observed with sufficient reliability.

## REFERENCES

- [1] 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.
- [2] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, “Scalable Application Layer Multicast,” in *Proc. SIGCOMM 2002*, Aug. 2002.
- [3] AD. A. Tran, K. A. Hua, and T. T. Do, “A Peer-to-Peer Architecture for Media Streaming,” in *IEEE JSAC: Special Issue on Advances in Overlay Networks*, Vol.22, N.1, Jan. 2004.
- [4] Y.-H. Chu, S. G. Rao, and H. Zang, “A Case for End System Multicast,” in *Proc. of ACM SIGMETRICS 2000*, June 2000.
- [5] X. Zhang, J. Liu, B. Li, and T. S. P. Yum, “DONet/CoolStreaming: A Data-driven Overlay Network for Live Media Streaming,” in *Proc. IEEE INFOCOM 2005*, Miami, Mar. 2005.
- [6] N. Magharei, R. Rejaie, “Understanding Mesh-based Peer-to-Peer Streaming,” in *Proc. NOSSDAV 2006*, Newport, Rhode Island, May 2006.
- [7] F. Pianese, J. Keller, and E. W. Biersack, “PULSE, a Flexible P2P Live Streaming System,” in *Proc. 9th IEEE Global Internet Symposium 2006*, Bascelona, Spain, Apr. 2006.
- [8] 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.
- [9] 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.
- [10] S. N. Dorogovtsev, and J. F. F. Mendes, “Evolution of Networks: From Biological Nets to the Internet and WWW,” Oxford University Press, Oxford, January 2003.
- [11] J. Leskovec, J. Kleinberg, and C. Faloutsos, “Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations,” in *Proc. 11th ACM SIGKDD 2005*, Chicago, IL, USA, Aug. 2005.
- [12] V. K. Goyal, “Multiple Description Coding: Compression Meets the Network,” in *IEEE Signal Processing Magazine*, pp. 7493, Sept. 2001.
- [13] M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava, “PROMISE: peer-to-peer media streaming using Collect-Cast,” in *Proc. ACM 2003*, Berkeley, CA, Aug. 2003.
- [14] H. P. Breuer and F. Petruccione “On the numerical integration of Burgers’ equation by stochastic simulation methods,” *Computer Physics Communications*, Vol. 77, Pages 207-218, 1993.
- [15] J. Honerkamp, “Stochastic Dynamical Systems: Concepts, Numerical Methods, Data Analysis,” 1994, VCH, New York.
- [16] D.T. Gillespie, “Exact Stochastic Simulation of Coupled Chemical Reactions,” *Journal of Physical Chemistry*, Vol. 63, Issue 25, Pages 2340-2361, 1977.
- [17] D. Carra, R. Lo Cigno, and E. W. Biersack, “Stochastic Graph Processes for Performance Evaluation of Content Delivery Applications in Overlay Networks,” submitted for publication. Available: [www.dit.unitn.it/locigno/preprints/CaLoBi.TPDS.V1.0.pdf](http://www.dit.unitn.it/locigno/preprints/CaLoBi.TPDS.V1.0.pdf)