Graph Based Modeling
of P2P Streaming Systems[*]

Damiano Carra[1], Renato Lo Cigno[1], and Ernst W. Biersack[2]

[1] Dip. di Informatica e Telecomunicazioni, Università di Trento, Trento, Italy
{carra,locigno}@dit.unitn.it
[2] Institut EURECOM, Sophia Antipolis, France
erbi@eurecom.fr

Abstract. This paper addresses the study of fundamental properties of stream-based content distribution services. We assume the presence of an overlay network 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 can capture the transient behavior of the distribution graphs, i.e., the evolution of the structure over time.

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.

1 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] coexist now with more advanced mesh-based systems [3][4][5] 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. Many proposals use heuristic methods to improve performance, but these heuristics are verified only 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., [6]) and meshes are studied only through simulation [5]. To the best of our knowledge, no study analyzes the behavior of the streaming distribution system as a function of the topological properties of the graph that is built by the P2P application. Many studies based on graph theory focus on the steady state properties of networks

[7][8], but the way the graph structure is built is not constrained by protocol rules.

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. Our model can analyze the *dynamics* of the graphs, i.e., the evolution of the structure over time. It can assess the impact of different protocol choices and of bandwidth heterogeneity on the streaming process, and it gives enough insight in the problem to formulate improved streaming strategies.

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 high churn, and the delay experimented by nodes is bounded.

2 Mesh-based Overlay Streaming Systems

We do not consider a specific system but we identify common basic characteristics of recent proposals [4][5]. Consider an overlay network built by a P2P application. Once the overlay layer is built, paths between the source and the destinations are created according to the rules of the streaming protocol. 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.

2.1 System parameters

The content is distributed using R different stripes. Each stripe contains part of the stream (coded, for instance, using MDC techniques [9]). A node needs $R' < R$ out of R stripes to achieve a target quality, while the remaining $R - R'$ stripes contain redundant information.

The evolution of the network is subject to two main events: node arrivals and node 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_{str} be the duration of the stream and N the mean number of nodes receiving the stream at steady state. A fraction of the nodes joins the stream at time zero, and there is an *arrival interval* during which $\lambda(t) > \mu(t)$ until steady state is reached. Fig. 1 shows a sample arrival pattern.

The departure rate $\mu(t)$ is the inverse of the mean time spent in the system (sojourn time). $\lambda(t)$ at steady state compensates departures. For a given time interval T, the ratio between the cumulative number of disappeared nodes and the mean number of active nodes during T is defined as the *churn* of the system.
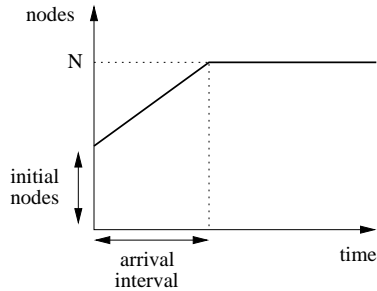
Fig. 1. Sample arrival pattern.

A 100% churn means that during T the number of the departed nodes is equal to the mean number of nodes in the system, i.e., there is a stochastic complete change of the nodes during T.

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_{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_{str}/R', and we assume that the server is able to upload all the R stripes, i.e., it has a bandwidth greater than $R r_{str}/R'$. Each node has a constraint on maximum and minimum number of active uploads that limit the possible outdegree of the node: k^{max} is the *maximum* and k^{min} is the *minimum outdegree*.

Each node has B neighbors. Among its neighbors the node selects its *parent* nodes, i.e., those from which it download. R' parents are called *active*; the remaining are called *standby*, since they are used as a backup in case of active parent failure.

2.2 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 with the help of the contacted node. From the neighbors list, the node selects its parents and attaches to them. Nodes periodically search for new connections among their neighbors with rate λ_{up} in order to increase their indegree.

When a node leaves, all the inbound and outbound connections are canceled. Orphan nodes try to replace the disappeared parent. If the disappeared parent

was in the standby set, the node does not react (it simply loses a backup parent). If the disappeared parent was in the active set, the node tries to switch the state of a standby parent, i.e., it starts downloading from the standby parent if it has available bandwidth. If the node has no backup parents, there is a temporary loss of quality that depends on the time necessary to search for a new parent.

3 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 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 focus of our analysis is the characteristics of the *distribution graph*, i.e., the subgraph of the overlay graph, where edges are the connections effectively used by nodes.

In general, the distribution graph 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 occurred during dt.

The distribution graphs can be described through their structural characteristics. We consider two main distributions: the degree distribution and the delay distribution [7].

The degree distribution $p_s(k, t)$ is the probability that node s has k connections at time t.[3] Knowing the degree distribution of each node in the graph, we can derive the total degree distribution

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

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 $p_s(\ell, t)$ as the probability that node s is ℓ steps away from the source at time t. The *total* delay distribution can be derived in a similar way as done for Eq. (1). Hereinafter, for notation simplicity, we omit the specification of node s.

3.1 Master Equations and Rate Equations

The analysis of the graph can be done through the study of the properties of the degree and delay distributions. For a Markov process, the temporal behavior can be described using the differential form of the Chapman-Kolmogorov equations, known as *Master Equations* (MEs) [7].

---

[3] It is possible to distinguish between indegree and outdegree distributions, $p_s(k_i, t)$ and $p_s(k_o, t)$ respectively, with $k_i + k_o = k$.

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

$$\frac{\partial}{\partial t} p(\alpha, t) = \sum_{\beta} w_{\beta,\alpha}(t) p(\beta, t) \tag{2}$$

where $w_{\beta,\alpha}(t)$ is the transition rate from the value β to the value α at time t. Transition rates are closely related to the streaming protocol policies and behavior. The general formulation of the MEs must be specialized for our problem, i.e., we have to define all the possible transitions.

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 $\overline{k}$ or delay $\overline{l}$). The correspondent equations are called *Rate Equations* (REs):

$$\frac{\partial}{\partial t} \overline{\alpha} = \frac{\partial}{\partial t} \sum_{\alpha} \alpha p(\alpha, t) \tag{3}$$

The REs 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 2 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 a cost of more resources necessary to find the solution), since they fully characterize the properties over time. REs gives a mean value that is equivalent to the fluid approximation of the system.
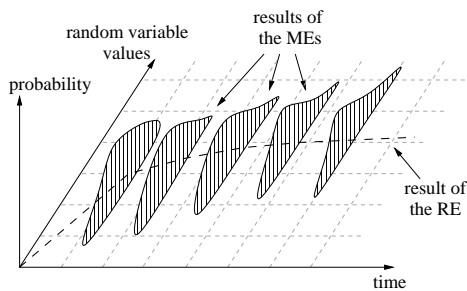

Fig. 2. Results of the Master Equations and the Rate Equations

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 method offers a great flexibility in deciding the desired level of detail in the system analysis.

Space forbids the detailed description of the transition rates: the interested reader can refer to [12].

4 Monte Carlo Integration of the MEs

The set of MEs that describe the distribution process 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 [10][11][4], 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. 2 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 can be estimated on the whole probability distribution.

The key strength of the methodology, however, is not 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 of the protocol/system. On the one hand, this is like a generic simulation approach, but, being based on formal definitions, avoids the risk of incomplete or bugged specifications; on the other hand, assumptions made in fluid models can be avoided, since we can describe the system behavior in full detail.

4.1 Comparison with Other Methodologies

We consider a very simple case in order to show the differences with other modeling approaches. Consider the case where a node update its indegree only during update events. We assume infinite upload and download bandwidths 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 $R - k_i(t)$ parents. In fact, under these assumptions the probability to find all the necessary parents 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

$$\frac{d}{dt}k_i(t) = \lambda_{\mathrm{up}}(R - k_i(t)) - k_i(t)\mu \tag{4}$$

The second term considers the fact that each of the $k_i(t)$ parents can leave with rate μ. Considering the initial condition $k_i(0) = 1$ (we suppose that all nodes are present at the beginning with exactly one parent each) the solution of (4) is

$$k_i(t) = \frac{\lambda_{\mathrm{up}}R}{\lambda_{\mathrm{up}} + \mu}\left(1 - e^{-(\lambda_{\mathrm{up}}+\mu)t}\right) + e^{-(\lambda_{\mathrm{up}}+\mu)t} \tag{5}$$

---

[4] 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.
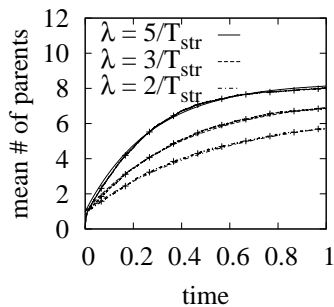
Fig. 3. Solution of MEs and REs.

In Fig. 3 we compare the analytical solution of this very simple case with the solution of the Rate Equations (3) derived from our model. We set $R = 10$ stripes, $\mu = 1/T_{\mathrm{str}}$ and $\lambda_{\mathrm{up}} = \frac{5}{T_{\mathrm{str}}}, \frac{3}{T_{\mathrm{str}}}$ and $\frac{2}{T_{\mathrm{str}}}$. We normalize the time with respect to T_{str}. The numerical solution follows closely the analytical one. But the results obtained from our model give more insight. In fact, we can observe how the *full* indegree distribution changes over time.
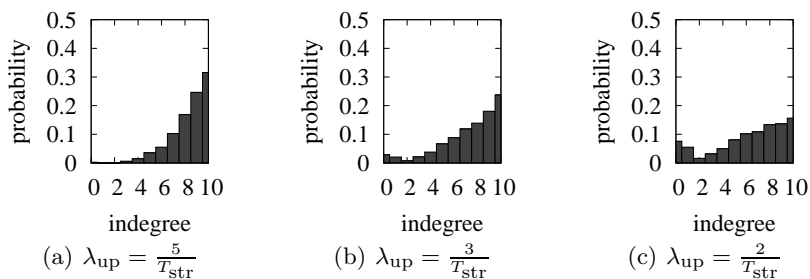

(a) $\lambda_{\mathrm{up}} = \frac{5}{T_{\mathrm{str}}}$ (b) $\lambda_{\mathrm{up}} = \frac{3}{T_{\mathrm{str}}}$ (c) $\lambda_{\mathrm{up}} = \frac{2}{T_{\mathrm{str}}}$

Fig. 4. Indegree distribution at time $T_{\mathrm{str}}/2$ obtained from the solution of the MEs.

Fig 4 shows the distribution of the number of parents (indegree) at time $T_{\mathrm{str}}/2$ for different values of λ_{up}. Notice that there is a non-null probability that nodes remain without parents, thus being disconnected entirely from the distribution process, a phenomenon that a fluid approach analyzing the means entirely disregards, while in most cases it is the most interesting result.

5 Application of the Methodology

We use a configuration with $N = 10^4$ nodes[5]. The input bandwidth distribution is composed by three classes: slow nodes, medium nodes and fast nodes with symmetric bandwidth respectively equal to r_{str}, $2r_{str}$ and $5r_{str}$ and probabilities equal to 0.2, 0.4 and 0.4. The streaming rate is divided into R' stripes and the source generates R stripes. Results are obtained for $R = 10$ and $R' = 2, 4, 6, 8$.

We consider an observation time equal to T_{str}. We consider two arrival patterns, with initial number of nodes equal to $0.1N$ and $0.5N$ respectively; the remaining nodes arrive within $T_{str}/5$. The mean sojourn time is set to T_{str}, $2T_{str}$, and $5T_{str}$. 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 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 $\frac{U}{r_{str}} = 1$ 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_{str}, is set to $10000\frac{U}{r_{str}} = 10000$ units.

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

Due to space constraints, we report only some sample results; a more comprehensive set can be found in [12]

5.1 Analysis of the Indegree

Analysing the indegree we can 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 5(a) shows the indegree distribution of the nodes at time $t = T_{str}$, computed with Eq. (1). In this case we have initial number of nodes equal to $0.1N$ and mean sojourn time T_{str}. The distribution tends to peak around R 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'. Note that with $R' = 8$ there is a fraction of the nodes with exactly 8 parents: this means that, in case of one parent disappears, the quality received by the node may be temporarily affected.

---

[5] We have also checked configurations with 10^5 nodes obtaining similar results.

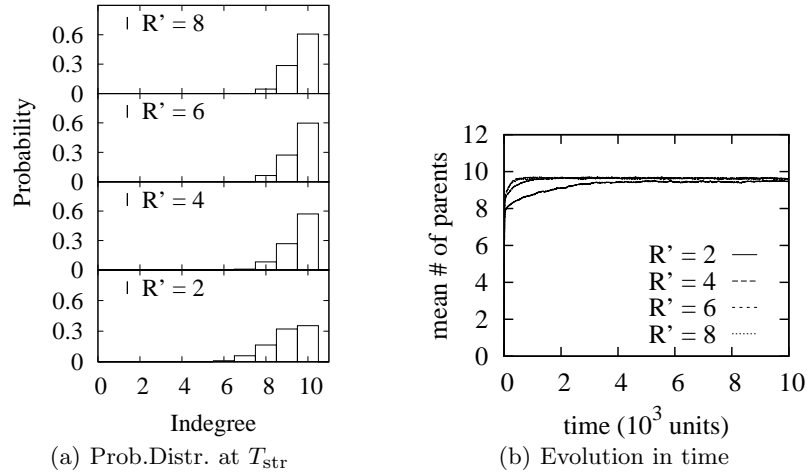(a) Prob.Distr. at T_{str} (b) Evolution in time

Fig. 5. Solution of the MEs for the indegree $k_i(t)$.

The average temporal behavior of the indegree can be analyzed looking at the results of the rate equations (Fig. 5(b)) computed with Eq. (3). 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.

5.2 Analysis of the Delay

The delay, expressed as time units, represents the number of hops from the source. We plot the Complementary Cumulative Distribution Function (CCDF, defined as 1-CDF) in order to study the tail of the distributions. We consider $R' = 4$ and we set different sojourn times (μ). Fig. 6(a) shows the case of initial number of nodes equal to $0.5N$. The tail of the distribution is not affected by the different values of μ.

In Fig. 6(b) we show the impact of R' on the delay. Increasing the number of stripes has a side effect: since each node needs 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.

5.3 Analysis of the Quality of the Mesh

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 1 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:
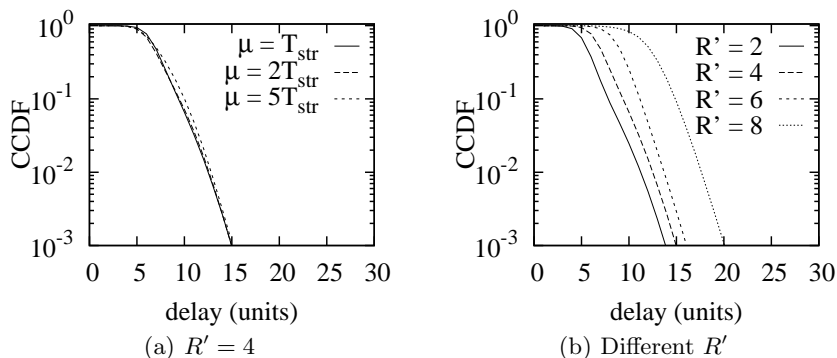
Fig. 6. CCDF of the delay (Initial number of nodes: $0.5N$)

arrivals and departures are Poisson processes with rate $\lambda(t)$ and $\mu(t)$ respectively, so we can calculate the cumulative disappeared nodes at time T_{str} and consequently the value of churn.

During node's lifetime, there is a non null probability that all its parents leave and node i is not able to find other parents. This 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 1 for different values of R'. With high churn, one node per thousand is forced out of the system. In some contexts, this value may be inacceptable.

Table 1. Other statistics

R'	$1/\mu$	% Churn	% Error	%Switch
4	T_{str}	98.33%	0.11%	99.63%
4	$2T_{\mathrm{str}}$	49.09%	0.05%	99.63%
4	$5T_{\mathrm{str}}$	19.77%	0.03%	99.63%
8	T_{str}	98.24%	0.10%	94.38%
8	$2T_{\mathrm{str}}$	49.15%	0.05%	96.76%
8	$5T_{\mathrm{str}}$	19.74%	0.02%	97.99%

Another interesting results is the probability to switch to a standby parent if an active parent 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 1). 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 parents of a node n drops below R', the probability to switch to a standby

parent is 94%. This means that the quality temporalily decreases, as expected looking at degree distribution (Fig. 5(a)).

6   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 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 [4] consider only a small fraction of redundant information so, in case of node departures, the stability of the streaming is affected. Maintaining standby parents, a common solution used by such systems, may not alleviate the problem, since a node can have a standby set of parents with the same stripes of the (remaining) active parents.

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.

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 disconnection rates are unacceptable and require simulating thousands of nodes for hours to be observed with sufficient reliability.

References

1. Pendarakis, D., Shi, S., Verma, D., Waldvogel, M.: ALMI: An Application Level Multicast Infrastructure. In *Proc. of the 3rd Usenix Symposium on Internet Technologies & Systems (USITS)* (2001)
2. Banerjee, S., Bhattacharjee, B., Kommareddy, C.: Scalable Application Layer Multicast. In *Proc. SIGCOMM* (2002)
3. Chu, Y.-H., Rao, S. G., Zang, H.: A Case for End System Multicast. In *Proc. of ACM SIGMETRICS* (2000)
4. Zhang, X., Liu, J., Li, B., Yum, T. S. P.: DONet/CoolStreaming: A Data-driven Overlay Network for Live Media Streaming. In *Proc. IEEE INFOCOM* (2005)
5. Magharei, N., Rejaie, R.: Understanding Mesh-based Peer-to-Peer Streaming. In *Proc. NOSSDAV* (2006)

6. Baccelli, F., Chaintreau, A, Liu, Z., Riabov, A., Sahu, S.: Scalability of Reliable Group Communication Using Overlays. In *Proc. IEEE INFOCOM* (2004)
7. Dorogovtsev, S. N., Mendes, and J. F. F.: Evolution of Networks: From Biological Nets to the Internet and WWW. Oxford University Press, Oxford (2003)
8. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations. In *Proc. 11th ACM SIGKDD* (2005)
9. Goyal, V. K.: Multiple Description Coding: Compression Meets the Network. IEEE Signal Processing Magazine, Vol. 18, Issue 5, pp. 74-93 (2001)
10. Honerkamp, J.: Stochastic Dynamical Systems: Concepts, Numerical Methods, Data Analysis. VCH, New York (1994)
11. Gillespie, D. T.: Exact Stochastic Simulation of Coupled Chemical Reactions. Journal of Physical Chemistry, Vol. 63, Issue 25, pp. 2340-2361 (1977)
12. Carra, D., Lo Cigno, R., Biersack, E. W.: "Graph Properties of Mesh-based Overlay Streaming Systems," TR DIT-06-043, Univ. of Trento (2006) Available: `http://www.dit.unitn.it/locigno/preprints/DIT-06-043.pdf`