Statistical Properties of Chunkless Peer-to-Peer Streaming Systems

Roberto Rinaldo and Riccardo Bernardini and Roberto Cesco Fabbro
DIEGM – University of Udine
Via delle Scienze 208, 33100 Udine - Italy
Email: {rinaldo, bernardini, roberto.cesco}@uniud.it

***Abstract*—Streaming solutions based on peer-to-peer networks have recently attracted the attention of the research community, due to the fact that the possibility to exploit peers' upload bandwidth can make it possible to transmit to a large number of users at a low costs. In this paper, we analyze the packet loss probability experienced at the application layer when a stream-based, *chunkless* peer-to-peer network is employed. More precisely, we derive a network model that allows to characterize the asymptotic behavior of the packet loss probability when the distance between a node and the server grows. Although in the limit the packet loss probability converges to 1, we derive analytical bounds on the convergence rate, which can be used to choose network parameters so that such probability remains negligible.**

I. Introduction

P2P networks are a promising approach for multimedia streaming to a large number of users because each user contributes to data propagation, making the system more scalable than non-P2P solutions, such as Content Delivery Networks. Because of this, P2P streaming recently attracted interest both in the research community and in industry [1]–[8].

Most works in the literature consider the case of a mesh network based on a *pull* approach [8]–[10]. P2P networks of this type are fairly common and are typically based on a BitTorrent-like approach: the multimedia content is split into *chunks* that are exchanged among peers. The other type of approach is instead more similar to multicast, and the data flow is *pushed* over an overlay network (typically, but not necessarily, organized as a multiple tree) [4]–[7], [11].

In order to make P2P a suitable solution for commercial streaming, it is important to be able to predict the different performance indicators (jitter, delay, erasure probability, achievable bandwidth, ...) in order to design efficient P2P networks that meet the user's expectations without waste of resources.

The problem of estimating the performance of P2P networks recently attracted the attention of the research community. Because of the number of possible approaches to P2P streaming, and of the number of possible performance indicators, literature about the performance of P2P networks is quite mixed. For example, some works consider the delay in a P2P networks and its impact on scalability and data loss [12], [13]; other works consider the effect of *churn* and channel changes [14]–[16].

Most of the works cited above are independent of the specific topology of P2P network or consider the case of *pull* networks. Among the works that specifically consider the case of *push* networks, [17] considers the case of a single transmission tree and analyzes the upload capacities of the peers. Multi-tree overlays were considered in [18] and in [19], [20] where it is shown that this type of network has a "phase transition" behavior if Forward Error Codes (FEC) are employed. Finally, the recent work [21] considers the effect of packet losses and node churn over the availability of packets at the network nodes. The network model considered in [21] is a multiple-tree overlay multicast, where packets are distributed in a round-robin fashion among the different trees.

In this paper, we are interested in the packet loss probability experienced by the application that "sees" the P2P network look like an erasure channel with an equivalent loss probability P_{eq}. The type of network that we analyze in this work is stream-based and *chunkless*, as considered in Octoshape [4], Lava [5], PPETP [11], R2 [6] and Split-stream [7]. The model that we analyze here is more general than the multiple-tree model. One key point is that it allows each node to lower its upload bandwidth by applying suitable *reduction procedures* to the multimedia content (see Sect. III for more details). The reduction procedure is employed in the schemes above to allow nodes with limited upload bandwidth to contribute to transmission, and in some case to increase the reliability of the system.

Among the papers cited above, [21] is closest to this work, although it uses a different network model and different performance metrics. More precisely, in [21], the authors consider the "global metric" represented by the probability that a randomly chosen node in the network can reconstruct a given packet, while we are interested in the asymptotic behavior of P_{eq} as a function of the distance between a node and the server. The main result that we show here is that, although the value of P_{eq} experienced by a node converges to 1 when the distance from the server grows to infinity, it is possible to choose network parameters that make this convergence very slow, so that P_{eq} remains negligible in networks of practical size. We show this by deriving analytical bounds on the P_{eq}

This work was carried out within the PRIN research project "ARACHNE: Advanced video streaming techniques for peer-to-peer networks," prot. 2008C59JNA, with the support of the Italian Ministry of University and Research (MIUR).

convergence rate.

This paper is organized as follows. In Section III we describe in some detail the model of the P2P system considered in this paper; in Section IV we introduce the network model; in Section V we analyze the asymptotic behavior of P_{eq}; in Section VI we report some simulation results. Finally, in Section VII we draw the conclusions.

II. Notation

If, in a P2P system of *push* type, node a sends data to node b, we will say that a is an *upper peer* of b and that b is a *lower peer* of a[1]. We will define $\delta_{a,b}$ as the *Kronecker delta*, that is, $\delta_{a,b} = 1$ if $a = b$ and $\delta_{a,b} = 0$ otherwise.

In this paper we will consider Markov chains with a finite alphabet. We will use $\rightarrow$ to denote a one-step reachability relation, that is, we will write $a \rightarrow b$ if the chain can transition from a to b in one step. We will use $a \rightarrow^n b$ if there is a path of length n from a to b and $a \rightarrow^* b$ if there is a path of *any* length from a to b. If the Markov chain is homogeneous, we will use the shorthand $P(a \rightarrow b_1 \rightarrow b_2 \rightarrow \cdots b_N)$ to denote $P[s_{n+N} = b_N, \ldots, s_{n+1} = b_1 | s_n = a]$. Note that this notation factorizes, that is, $P(a \rightarrow b \rightarrow c) = P(a \rightarrow b)P(b \rightarrow c)$.

III. Abstract P2P Streaming System

A. Reduction procedures

The P2P streaming system considered in this paper is an abstract system that generalizes the behavior of many existing P2P streaming systems of *push* type such as like Octoshape [4], Lava [5], R2 [6] and Split-stream [7] and PPETP [11]. An important characteristic of most of these schemes is that the streams produced by the nodes are not copies of the whole content stream, but they are instead *reduced streams* that require a fraction of the bandwidth of the content stream. This approach has several advantages, the most obvious one being the fact that in this way even nodes with small upload bandwidth can contribute to data propagation. Other advantages are a greater reliability of the system and a protection from some attacks such as the *stream poisoning attack* [22], [23].

The details of how the reduced streams are produced are not important for the scope of this work. We will describe the abstract reduction procedure by supposing that the P2P streaming scheme defines a set of *reduction functions*[2] $\{\phi_\mu\}_{\mu \in \mathscr{M}}$, indexed by a set $\mathscr{M}$ of *reduction parameters*. Each node selects a reduction parameter $\mu \in \mathscr{M}$ and processes each content packet c with the corresponding reduction function to obtain the *reduced versions* $r_\mu = \phi_\mu(c)$ that are forwarded to the lower peers. The size of r_μ is, typically, a fraction of the size of c. In this paper we will suppose, for the sake of concreteness, that the size of r_μ is $1/R$ times the size of c.

In this paper, we are not interested in the details of the definition of ϕ_μ. It suffices that the set $\{\phi_\mu\}_{\mu \in \mathscr{M}}$ satisfies the following *R-reconstruction* hypothesis.

[1]Therefore, data flow from top to bottom.

[2]Function ϕ_μ typically is a linear combination (in a finite field) of the data in c, but this is not necessary.

Hypothesis 1 (R-reconstruction)**.** *Content packet c can be recovered from the knowledge of* any *set of R different reduced versions $r_{\mu_1} = \phi_{\mu_1}(c), \ldots, r_{\mu_R} = \phi_{\mu_R}(c)$.*

Remark III.1

The easiest way to satisfy the R-reconstruction property is by using Reed-Solomon codes [4], [11], [7], but other solutions are possible [24].

B. Node behavior

Supposing that the R-reconstruction hypothesis holds, the typical node behavior in our model is the following

1) When a node starts, it chooses a reduction parameter $\mu \in \mathscr{M}$. Moreover, the node contacts $N \geq R$ upper peers.
2) As soon as the node receives at least R reduced different reduced versions of c, it recovers c and moves it to the application level. Moreover, the node processes c with ϕ_μ and forwards the result to its lower peers.

Remark III.2

Few remarks are in order

1) Note that each node *does not* forward the received reduced packets (see Sect. III-C for an exception), but it *regenerates* c before creating a new reduced version which is sent to the lower peers.
2) If $N > R$ the node receives a redundant set of data. This can be exploited to make the system robust with respect to packet losses, *churn* and poisoning attacks [22].
3) The reduction parameters chosen by the N upper peers must be different from one another. This can be granted by assigning them in a centralized way, but if $\mathscr{M}$ is large enough, peers can choose their parameter at random and still have a small probability of duplicated parameters.
4) The abstract system described here can be adapted also to other schemes such as the round-robin scheme in [21]. Toward such an end, one can consider c as a "macro packet" collecting τ consecutive packets, and defining τ reduction functions $\phi_1, \ldots, \phi_\tau$, where ϕ_μ "extracts" the μ-th packet from the macro-packet c.

C. Fragment Propagation

It could happen that, because of packet losses, the node receives less than R reduced versions of c. In this case the node cannot recover c, but can nevertheless help in propagating the information about c, by forwarding to its lower peers one of the reduced packets received by its upper peers. If this happens, we will say that the P2P scheme employs *fragment propagation*.

IV. Network model

The P2P network will be represented by a Direct Acyclic Graph (DAG) where edges link each node to its lower peers. The server(s) will be clearly represented by node(s) that do not have upper peers. If more than one server is present, we suppose that they are organized as a Content Distribution Network (CDN) and feed their lower peers with the reduced versions of the same content packet. For the sake of notational simplicity, we will suppose that every node has $N \geq R$ upper peers and that every link is an erasure channel that drops packets with probability P_ℓ.

We associate with each node m of the network the random variable W_m defined by the following experiment. We let the server(s) send to the network a single content packet, we count the number of packets received by node m and we let $W_m \in \{0,1,\dots,N\}$ be such a number. From the knowledge of the statistical properties of W_m, it is possible to determine several values of interest. For example, the packet loss probability P_{eq} seen by the application can be computed as $P_{\text{eq}} = P[W_m < R]$.

As explained in Sect. III-C, a node sends reduced packets to its lower peers if it receives at least T reduced packets, where $T = 1$ if fragment propagation is employed and $T = R$ otherwise. If node n received at least T reduced packets (i.e., if $W_n \geq T$) we will say that the node is *active* or in *firing state*. We will define the random variable F_n to be equal to 1 if node n is in firing state and 0 otherwise.

A. Limited Spread Networks

A difficulty in studying the behavior of the abstract P2P system considered here is that the statistical properties of W_n depend on the network topology, a characteristic that it is not easily captured by a small set of parameters. In order to simplify the study, it is convenient to put some constraint on the topology.

A useful constraint that nevertheless is general enough to describe practical networks is the hypothesis of *limited spread*.

Definition 1. *If n is a node of the network define $d(n)$ and $D(n) \geq d(n)$ as the length of the shortest and longest path from the server to n. Value $D(n)$ will be called the* depth *of node n, and difference $D(n) - d(n)$ will be called the* spread *of n.*

Definition 2. *A network $\mathscr{N}$ will be said to be a* Δ-limited spread *network if $D(n) - d(n) \leq \Delta$ for every node $n \in \mathscr{N}$. A network will be said* stratified[3] *if $D(n) - d(n) = 0$ for every n. If a network $\mathscr{N}$ is stratified, define the K-th stratum S_K as*

$$S_K := \{n \in \mathscr{N} : D(n) = K\} \tag{1}$$

Remark IV.1

The hypothesis of limited spread is quite natural and it is expected that this type of networks will be the natural outcome of the tentative of maximizing locality. It is easy to prove that in a stratified network the upper peers of nodes in S_K belong to S_{K-1}.

1) Notation for stratified networks: We will denote with L_K the number of nodes in stratum K. The n-th node in stratum K, $n = 0,\dots,L_K - 1$, will be named as (K,n). The set of upper peers of (K,n) will be represented by the vector $\mathbf{u}_{K,n} \in \{0,1\}^{L_{K-1}}$ whose m-th component is 1 if $(K-1,m)$ is an upper peer of (K,n) and zero otherwise.

We will collect all the random variables $W_{K,n}$ and $F_{K,n}$, relative to nodes of stratum K, in two vectors $\mathbf{W}_K$ and $\mathbf{F}_K$; more precisely, $[\mathbf{W}_K]_n = W_{K,n}$ and $[\mathbf{F}_K]_n = F_{K,n}$.
It will prove useful to have a special notation for some states in $\{0,1\}^{L_K}$. More precisely, we will define the *empty state* as $\phi = [0,0,\dots,0]$ (no node in active state), the *full state* as $\Omega = [1,1,\dots,1]$ (every node in active state) and, for every $k \in \{0,\dots,L_K - 1\}$, the *k-th singleton state*, $\mathbf{e}_k$ as $[\mathbf{e}_k]_n = \delta_{k,n}$ (only the k-th node is active).

Finally, we define a *constant geometry* as a stratified network such that every stratum has the same number of nodes (denoted as L in the following) and $\mathbf{u}_{K,n}$ depends only on n, that is, $\mathbf{u}_{K,n} = \mathbf{u}_{M,n}$ for every $M,K \in \mathbb{N}$ and $n \in \{0,\dots,L-1\}$.

V. Asymptotic Analysis

For the sake of notational simplicity, in this section we suppose that the number of nodes per stratum is constant and drop the subscript from L_K.

A. Reduction to the analysis of $\{\mathbf{F}_K\}_{K\in\mathbb{N}}$

As anticipated, we are interested in the asymptotic behavior of variables $W_{K,n}$ when K goes to infinity. It is clear that, in a stratified network, the random variable sequence $\{\mathbf{W}_K\}_{K\in\mathbb{N}}$ is a Markov chain with alphabet $\{0,1,\dots,N\}^L$ and that the chain is homogeneous if the network has *constant geometry*. It is immediate to check that the transition probability is[4]

$$P[W_{K,n} = a | \mathbf{F}_K = \mathbf{s}] = P[\mathscr{B}(\mathbf{s}^t \mathbf{u}_{K,n}, 1 - P_\ell) = a] \tag{2}$$

where $\mathscr{B}(M,p)$ is a binomial random variable with M trials and success probability $1 - P_\ell$. According to (2), the transition probability between two different states in $\{0,1,\dots,N\}^L$ depends only on the pattern of active nodes at stratum $K-1$ and not on the actual number of received packets. By exploiting (2), it is possible to show that the sequence of random vectors $\{\mathbf{F}_K\}_{K\in\mathbb{N}}$ is also a Markov chain. Note that, because of (2), it suffices to study the statistical behavior of the chain $\{\mathbf{F}_K\}_{K\in\mathbb{N}}$.

Let $\mathbf{M}$ be the matrix of transition probabilities

$$\mathbf{M}_{r,c} := P(r \to c) = P[\mathbf{F}_K = c | \mathbf{F}_{K-1} = r] \tag{3}$$

and let λ_i be the eigenvalues of $\mathbf{M}$, ordered by decreasing modulus, that is, $|\lambda_1| \geq |\lambda_2| \geq \dots$.

B. Asymptotic behavior of $\{\mathbf{F}_K\}_{K\in\mathbb{N}}$

A first, almost obvious, but important result is the following.

Property 1. *The steady state probability of $\{\mathbf{F}_K\}_{K\in\mathbb{N}}$ is*

$$\lim_{K\to\infty} P[\mathbf{W}_K = \mathbf{s}] = \delta_{\mathbf{s},\phi}, \tag{4}$$

that is, the state of $\{\mathbf{F}_K\}_{K\in\mathbb{N}}$ will eventually converge to the empty state.

Property 1 is a consequence of the fact that ϕ is *absorbing*.

Equation (4) could seem as bad news for streaming over P2P networks, since it claims that nodes that are "very far" from the server will not receive any packets. In order to make precise what "very far" means, it is important to study λ_2 that, as well known, controls the velocity of convergence of (4). In order to state the main result about λ_2, we need a generalization of the concept of absorbing state.

[3]We use the term *stratified* to avoid confusion with the term *layered* possibly used in other contexts.

[4]Note that $\mathbf{s}^t \mathbf{u}_{K,n}$ is equal to the number of upper peers of (K,n) in active state.

Definition 3. *Consider a Markov chain with alphabet A and absorbing state $\phi \in A$. A state $s \in A$ is said to be a* trapdoor state *if there is an integer $\mathscr{L}_s$ such that $P(s \to^{\mathscr{L}_s} \phi) = 1$.*

Note that an absorbing state is a trapdoor and that a trapdoor state can transition only to another trapdoor state.

The following theorem gives upper and lower bounds on λ_2.

Theorem 1. *Consider the case of a constant geometry network with transition matrix $\mathbf{M}$. Let $\mathscr{T}$ be the set of trapdoor states. If $\mathbf{a} \to^* \Omega$ for every $\mathbf{a} \notin \mathscr{T}$, then $\lambda_2 \in \mathbb{R}$, $\lambda_2 > |\lambda_3|$ and*

$$P(\Omega \to \Omega) \leq \lambda_2 \leq 1 - P(\Omega \to \phi) \tag{5}$$

Proof: For the sake of space, some easy details are skipped. Transition matrix $\mathbf{M}$ can be written in the form

$$\mathbf{M} = \left[\begin{array}{c|c|c} 1 & 0 & 0 \\ \hline * & \mathbf{H} & 0 \\ \hline * & * & \mathbf{Q} \end{array}\right] \tag{6}$$

where the first row (and column) corresponds to the absorbing state ϕ, the second block of rows (and columns) correspond to the trapdoor states and the third block to non-trapdoor states.

It is clear that $\lambda_1 = 1$ and that the remaining eigenvalues of $\mathbf{M}$ are distributed between the eigenvalues of $\mathbf{Q}$ and the eigenvalues of $\mathbf{H}$. By definition of trapdoor state, it follows that all the non-zero eigenvalues of $\mathbf{M}$ are eigenvalues of $\mathbf{Q}$.

In order to prove that λ_2 is strictly dominating, one shows that $\mathbf{Q}$ is irreducible and primitive by observing that (i) for every $\mathbf{a}, \mathbf{b} \notin \mathscr{T}$ one has $\mathbf{a} \to^* \Omega \to \mathbf{b}$, and (ii) $P(\Omega \to \Omega) \neq 0$.

The lower bound in (5) follows by the fact that the spectral radius of a non-negative matrix is never smaller than the diagonal elements. The upper bound in (5) follows from the fact that in a non-negative matrix the spectral radius is not larger than the maximum row sum, that is

$$\lambda_2 \leq \max_{r \in \mathscr{T}^c} \sum_{c \in \mathscr{T}^c} P(r \to c) = \max_{r \in \mathscr{T}} 1 - P(r \to \mathscr{T}) \tag{7}$$

Since ϕ is a trapdoor state, $P(r \to \mathscr{T}) \geq P(r \to \phi)$ and one deduces

$$\lambda_2 \leq \max_{r \in \mathscr{T}} 1 - P(r \to \phi) = 1 - P(\Omega \to \phi) \tag{8}$$

where the last equality follows from the fact that $P(r \to \phi)$ is minimized when $r = \Omega$. ■

It is worth to write explicitly the probabilities in (5)

$$P(\Omega \to \Omega) = P[\mathscr{B}(N, 1 - P_\ell) \geq T]^L \tag{9a}$$
$$P(\Omega \to \phi) = P[\mathscr{B}(N, 1 - P_\ell) < T]^L \tag{9b}$$

that in the case of fragment propagation become

$$P(\Omega \to \Omega) = (1 - P_\ell^N)^L \approx 1 - LP_\ell^N \tag{10a}$$
$$P(\Omega \to \phi) = P_\ell^{NL} \tag{10b}$$

Example V.1

A simple numerical example can help understanding the meaning of Theorem 1. Consider the case of fragment propagation, $P_\ell = 0.1$, $N = 8$ upper peers per node and $L = 100$ nodes per stratum. According to (5) and (10a), λ_2 is not smaller than

$$P(\Omega \to \Omega) \approx 1 - 100 \cdot 0.1^8 = 1 - 10^{-6} \tag{11}$$

It is easy to verify that in order to have $\lambda_2^K < 0.99$ it is necessary to have $K > 10^4$. This shows that although "very far" nodes will receive very few packets, the convergence is very slow and it is quite unlikely that one will find "very far" nodes in practical contexts.

In the case of no fragment propagation and $R = 5$, the lower bound of λ_2 is

$$P(\Omega \to \Omega) = P[\mathscr{B}(8, 0.9) \geq 5]^{100} \approx 0.6 \tag{12}$$

which is much smaller than (10). Although this is only a lower bound, it suggests that convergence to the empty state can be very fast if fragment propagation is not used.

Remark V.1 (Extension to non-constant geometry and non-stratified networks)

Note that the bounds in (5) do not depend on the connections between consecutive layers and this suggests that a similar result can hold also in the case of non-constant geometry networks.

Moreover, if $P(\Omega \to \Omega)$ is large enough, the decay is so slow that, intuitively, it should not make much difference if node at stratum K receives its data from layer $K - 1$ or $K - \Delta$, as long as Δ is not too large. This suggests that, at least in the $P(\Omega \to \Omega) \approx 1$ case, similar results could also hold for non stratified networks. This claim is supported by the experimental results given in Sect. VI

VI. Simulation results

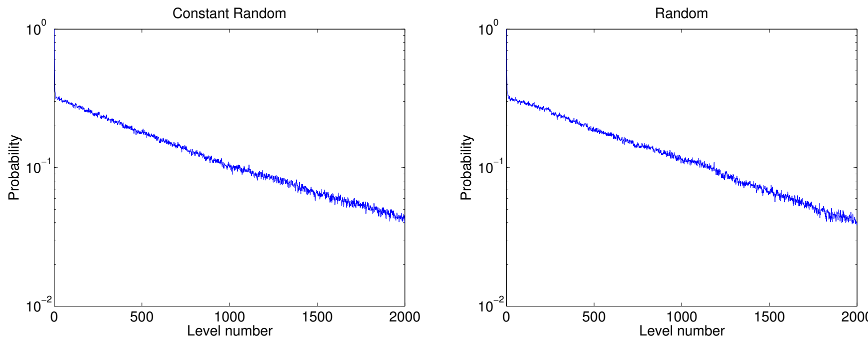
We carried out some simulations in order to complement the analytical results. We carried out the simulations using two different types of networks:

- A *constant random network*, that is a constant geometry networks where vectors $\mathbf{u}_{1,1}, \ldots, \mathbf{u}_{1,L}$ are independently drawn from the set U_N of vectors in $\{0,1\}^L$ with N entries equal to one;
- A *totally random networks* where every $\mathbf{u}_{K,j}$, is a randomly drawn from U_N.

For every network we simulated packet propagation across strata 100 times and we measured the frequency of packet reception at each node. In the case of random networks (i.e., constant random and totally random networks) the above procedure was repeated 20 times, with randomly chosen link layouts, and the results were averaged.

Plots in Fig. 1 show the measured probability of receiving at least one packet the two networks used in the experiments. In every case we chose $L = 10$, $N = 3$ and $P_\ell = 0.5$. (We chose such a large value of P_ℓ in order to have the probability decay visible.)

Note in Fig. 1a the exponential decay predicted by the theory above; note, however, that the same exponential decay happens also in the case of Fig. 1b, corresponding to a non-constant geometry network, supporting the claim in Remark V.1. Note that the networks exhibit very similar decay rate.



(a) (b)

Figure 1. Probability of packet reception in (a) a *constant random* network and (b) a *totally random* network. In all the plots $L = 10$, $N = 3$ and $P_\ell = 0.5$.

VII. Conclusion

We analyzed the packet loss probability P_{eq} experienced by the application when a stream-based, *chunkless* P2P network is employed. We show that, although in the limit P_{eq} converges to 1, it is possible to choose the network parameters to make this convergence so slow that such probability remains negligible in networks of practical size. Moreover, such a convergence can be made as slow as desired without increasing the redundancy in the network.

References

[1] F. Pianese, "A survey of p2p data-drivern live streaming systems," in *Streaming Media Architectures, Techniques, and Applications*, C. Zhu, Y. Li, and X. Niu, Eds. Hershey, NY, USA: Information Science Reference, 2011, ch. 12, pp. 295–310.

[2] D. Perino and F. Mathieu, "Epidemic live streaming," in *Streaming Media Architectures, Techniques, and Applications*, C. Zhu, Y. Li, and X. Niu, Eds. Hershey, NY, USA: Information Science Reference, 2011, ch. 13, pp. 311–336.

[3] Z. Liu, Y. Shen, K. W. Ross, S. S. Panwar, and Y. Wang, "Layerp2p: using layered video chunks in p2p live streaming," *Trans. Multi.*, vol. 11, pp. 1340–1352, November 2009. [Online]. Available: http://dx.doi.org/10.1109/TMM.2009.2030656

[4] S. Alstrup and T. Rauhe, "Introducing Octoshape – a new technology for large-scale streaming over the Internet," *EBU Technical Review*, no. 303, Jul. 2005.

[5] M. Wang and B. Li, "Network coding in live peer-to-peer streaming," *IEEE Transactions on Multimedia*, vol. 9, no. 8, pp. 1554–1567, Dec. 2007.

[6] ——, "R2: Random push with random network coding in live peer-to-peer streaming," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1655–1666, December 2007.

[7] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "Splitstream: High-bandwidth multicast in cooperative environments," in *19th ACM Symposium on Operating Systems Principles, 2003*, 2003. [Online]. Available: citeseer.ist.psu.edu/article/castro03splitstream.html

[8] X. Zhang, J. Liuy, B. Liz, and P. Yum, "CoolStreaming/DONet: a data-driven overlay network for efficient live media streaming," in *Proceedings of IEEE International Conference on Computer Communications*. IEEE Computer Society, Mar. 2005.

[9] "Pplive," http://www.pplive.com.

[10] "Ppstream," http://www.ppstream.com.

[11] R. Bernardini, R. C. Fabbro, and R. Rinaldo, "Peer-to-peer epi-transport protocol," http://tools.ietf.org/html/draft-bernardini-ppetp, Jan. 2011, internet Draft, work in progress.

[12] I. Chatzidrossos, G. Dn, and V. Fodor, "Delay and playout probability trade-off in mesh-based peer-to-peer streaming with delayed buffer map updates," *Peer-to-peer Networking and Applications (PPNA)*, vol. 3, Mar. 2010.

[13] G. Dn and V. Fodor, "Delay asymptotics and scalability for peer-to-peer live streaming," *IEEE Trans. on Parallel and Distributed Systems*, vol. 20, pp. 1499–1511, Oct. 2009.

[14] D. Wu, Y. Liu, and K. W. Ross, "Modeling and analysis of multichannel p2p live video systems," *IEEE/ACM Trans. Netw.*, vol. 18, pp. 1248–1260, August 2010. [Online]. Available: http://dx.doi.org/10.1109/TNET.2009.2038910

[15] D. Wu, Y. Liu, and K. Ross, "Queuing network models for multi-channel p2p live streaming systems," in *INFOCOM 2009, IEEE*, april 2009, pp. 73 –81.

[16] R. Kumar, Y. Liu, and K. Ross, "Stochastic Fluid Theory for P2P Streaming Systems," in *Proceedings of IEEE Infocom*, 2007.

[17] T. Small, B. Liang, and B. Li, "Scaling laws and tradeoffs in peer-to-peer live multimedia streaming," in *Proceedings of the 14th annual ACM international conference on Multimedia*, ser. MULTIMEDIA '06. New York, NY, USA: ACM, 2006, pp. 539–548. [Online]. Available: http://doi.acm.org/10.1145/1180639.1180754

[18] G. Dn, V. Fodor, and I. Chatzidrossos, "On the performance of multiple-tree-based peer-to-peer live streaming," in *Proc. of IEEE Infocom*, May 2007.

[19] G. Dn, V. Fodor, and G. Karlsson, "On the stability of end-point-based multimedia streaming," in *Proc. of IFIP Networking 2006*, May 2006.

[20] G. Dn, V. Fodor, and I. Chatzidrossos, "Streaming performance in multiple-tree-based overlays," in *Proc. of IFIP Networking 2007*, May 2007.

[21] G. Dn and V. Fodor, "Stability and performance of overlay multicast systems employing forward error correction," *Performance Evaluation*, vol. 67, pp. 80–101, Feb. 2010.

[22] R. Bernardini, R. Rinaldo, and A. Vitali, "A reliable chunkless peer-to-peer architecture for multimedia streaming," in *Proc. Data Compr. Conf.*, Brandeis University. Snowbird, Utah: IEEE Computer Society, Mar. 2008, pp. 242–251.

[23] X. Hei, Y. Liu, and K. W. Ross, "IPTV over P2P streaming networks: The mesh-pull approach," *IEEE Communications Magazine*, vol. 46, no. 2, pp. 86–92, Feb. 2008.

[24] R. Bernardini, R. C. Fabbro, and R. Rinaldo, "Group based reduction schemes for streaming applications," *ISRN Communications and Networking*, 2011, to be published.