

# Systematic Procedure for Improving Continuity and Latency on a P2P Streaming Protocol

María Elisa Bertinat, Darío Padula,  
Franco Robledo Amoza and Pablo Romero  
Laboratorio de Probabilidad y Estadística,  
Facultad de Ingeniería, Universidad de la República  
Julio Herrera y Reissig 565,  
11300, Montevideo, Uruguay,

Daniel De Vera and Pablo Rodríguez-Bocca  
Departamento de Investigación Operativa,  
Facultad de Ingeniería,  
Universidad de la República  
Julio Herrera y Reissig 565,  
11300, Montevideo, Uruguay,

**Abstract**—This work proposes a new piece selection strategy for improving latency and continuity in a P2P streaming network. Different piece selection strategies based on a simple and symmetrical model for sharing are considered. It is well known the scalability properties of Rarest First and low values of latency of Greedy. In this work these techniques have been revised, and a new family of strategies is proposed. The richness of this family is expressed in the Approximation Strategy Property, which shows that the shape of every feasible strategy can be approximated by one member of this family proposed. It is introduced a Permutation Finder Algorithm, which looks for strategies inside this family that achieve a tradeoff between continuity and latency, not found in previous related works.

**Keywords:** P2P; Rarest First; Greedy; Performance; Piece Selection Strategy.

## I. INTRODUCTION

Internet-based multimedia systems have many different architectures, depending on their sizes and on the popularity of their contents. The majority of them have a traditional CDN (Content Delivery Network) structure [1], [2], where a set of datacenters absorbs all the load, that is, concentrates the task of distributing the content to the customers. This is, for instance, the case of msnTV, YouTube, Jumpty, etc., all working with video content.

Another popular alternative consists of using the often idle capacity of clients to share the video distribution with servers, through the present mature Peer to Peer (P2P) systems [3], [4], [5]. These are virtual networks developed at the application level over the Internet infrastructure. The nodes in the network, called peers, offer their resources to the other nodes, basically because they all share common interests. As a consequence, as the number of customers increases, the same happens with the global resources of the P2P network. Nowadays P2P networks play an important role because of its popularity and impact on Internet traffic. Some commercial P2P networks for live video distribution are available, all of them with proprietary source-codes and protocols. The most successful are PPlive, SopCast, PPstream, TVAnts and TVUnetwork. This work is part of the GoalBit project [6], [7], the first open-source P2P streaming network.

The dynamism and freedom of peers in a P2P network makes on one hand an attractive and powerful tool for them,

and on the other imposes many challenges in architecture design and protocols for sharing information [8], [9]. Live video P2P networks have harder constraints to satisfy, because many nodes only remain connected a few minutes [10]. Surveys about P2P networks for live video distribution [11], [12], [13] show that the continuity in the reproduction and the delay of the video are the most important factors in the quality of experience perceived by end users. See studies [14], [15] for details. A high cooperation between peers, and specially an efficient piece selection strategy are indispensable in order to obtain high reproduction continuity and low latency in a P2P streaming network [16]. This work extends the study done in [17]. We present a new proposal of piece selection strategy that has better results than previously considered strategies.

This paper is organized as follows. In Section II a simple model for P2P stream sharing is presented. In Section III previously piece selection strategies are revised. Section IV shows our new family of piece selection strategies and an algorithm which allows to find the better ones, defining optimality in a mathematical sense. A comparison between classical strategies and a new one is also shown. Finally in Section V the main conclusions of this work are presented.

## II. SIMPLE MODEL FOR SHARING

Let us consider a P2P network of  $M$  identical peers, each one with a buffer of size  $N$ , and a server which has the entire video content. Time is slotted and the server chooses in each slot one peer at random to send the present video piece. Peers interchange the pieces with some cooperation strategy. It is assumed that every peer can select another one so as to make a request and possibly, obtain a new piece (or not), all in less time than one time slot. They are responsible to assemble these pieces so as to obtain the desired video. Each peer displays the oldest piece in its buffer (i.e. the piece at position  $N$ ). A distortion is perceived by the user if he could not obtain this piece in time. See Figure 1 for a graphical description.

The request consists of contacting one peer and ask if in a given index of its buffer (which is empty or corrupted in the original peer) it has the correct piece. If this is the case, the peer may get this piece and the request is successful.

Let us call  $p_i$  to the probability that a peer has the correct piece in the  $i^{th}$  buffer position. Consider that a particular peer  $k$  selected peer  $h$  to download a piece. Using a piece selection strategy, suppose that the buffer at index  $i$  is owned by peer  $h$  and desirable for peer  $k$ . Let us call this event  $S(h, k, i)$  with probability  $s_i$ . It can be demonstrated [17] that under these conditions, and assuming stationarity, the probability of having correct buffers at  $i$ ,  $p_i$ , is a monotonous function in the playback direction. More explicitly:

$$p_1 = 1/M; \quad (1)$$

$$p_{i+1} = p_i + (1 - p_i)p_i s_i, \quad i = 1, \dots, N - 1 \quad (2)$$

It is suggested to revise [17] for more details.

The continuity of playback is measured by  $C = p_N$ , i.e. the stationary probability of having the next piece to be reproduced. The measure of the start up latency has some issues involved. Starting with an empty buffer and knowing that a peer obtains at most one piece for each time slot, it will need  $\sum_{i=1}^N p_i$  time slots to obtain the number of pieces of the stationary state. We assume that the P2P application starts the reproduction when the peer reaches the stationary state. If it starts early more time will be needed to reach the stationary state and more discontinuities will be in the playback reproduction during this period. Otherwise, there is not reason to start the reproduction later. Therefore, the start up latency of the video streaming is measured by:

$$L = \sum_{i=1}^N p_i.$$

This work is focused on defining a new piece selection strategy with high continuity, and at the same time low latency, which is presented in Section IV. Before, let us revise some classical strategies and its performances.

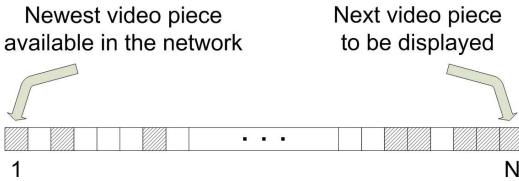


Fig. 1. Buffer model at each peer. Position 1 represents the newest video piece in the network, and  $N$  represents the next piece to be displayed. Observe that all peers are synchronized in the buffer consumption

### III. CLASSICAL STRATEGIES AND A MIXTURE

This section presents two classical piece selection strategies, named Rarest First and Greedy, and a mixture of them, attending to its performances. Rarest First enjoys a prestigious place nowadays, original by BitTorrent [18]. With a Rarest First strategy the peer chooses to download the pieces that are locally rarest, in order to guarantee a high piece diversity. In a streaming context, it basically consists on selecting a missing

piece that the contacted peer has, looking initially far away from the playback (rare, because of monotonicity of  $p$ ):

$$s_i = \left(1 - \frac{1}{M}\right) \prod_{j=1}^{i-1} (p_j + (1 - p_j)^2) \quad (3)$$

It has a clear interpretation: buffer  $i$  will be chosen if the local peer was not selected by the server, does not have the piece and the contacted peer does. Moreover, all previous buffers fall in two cases: the local peer already owned that piece or it does not but the contacted peer neither.

By the other side, it is well known that Greedy achieves better latency, but it is not scalable as Rarest First [19], [17]. The Greedy strategy is identical to the previous one, but looks first for the pieces nearest to the playback:

$$s_i = \left(1 - \frac{1}{M}\right) \prod_{j=i+1}^{N-1} (p_j + (1 - p_j)^2) \quad (4)$$

Figure 2 shows graphically the Rarest First and Greedy strategies.

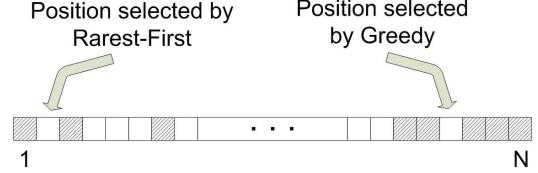


Fig. 2. Rarest First and Greedy buffer strategies

A mixture of this latter strategies is possible, cutting the buffer in a given index  $m$ :  $1 \leq m \leq N$  and applying the Rarest First strategy in the first partition and Greedy in the other [17]. This strategy offers a more reduced latency than Rarest First and good continuity.

### IV. A NEW STRATEGY

In this work it is proposed a systematic procedure for improving latency and continuity, which achieves a tradeoff not found in previously considered strategies. Let us consider a permutation of the first  $N - 1$  buffer positions:

$$\pi : \{1, \dots, N - 1\} \rightarrow \{1, \dots, N - 1\}, \\ \forall i \neq j, \pi(i) \neq \pi(j).$$

Then, let us define the family of all strategies (one per each permutation):

$$s_{\pi(i)} = \left(1 - \frac{1}{M}\right) \prod_{j=1}^{i-1} (p_{\pi(j)} + (1 - p_{\pi(j)})^2) \quad (5)$$

The interpretation is similar to the classical ones, excepting that the order of query is different: first, piece at position  $\pi(1)$  is requested. If the local peer has it, or it does not and the selected peer neither, the local peer asks next for buffer  $\pi(2)$ , and so on. This family of strategies enjoys at least two very interesting properties, which guide us to define a systematic procedure for improving the performance of other strategies.

**Property IV.1.**  $s_{\pi(i)}$  is a strictly monotonous decreasing sequence.

*Proof:* By its definition, all permutation strategies satisfy the next recursion:

$$s_{\pi(1)} = 1 - \frac{1}{M} \quad (6)$$

$$s_{\pi(i+1)} = s_{\pi(i)}(p_{\pi(i)} + (1 - p_{\pi(i)})^2) \quad (7)$$

$$\forall i \in 1, \dots, N-2$$

Moreover, it is nonnegative because it is obtained by product and sums of nonnegative factors and terms. It is easy to prove by induction that  $p_i \in [\frac{1}{M}, 1], \forall i = 1, \dots, N$ . Then, we have the next relation:

$$s_{\pi(i+1)} - s_{\pi(i)} = s_{\pi(i)}(p_{\pi(i)}(p_{\pi(i)} - 1)) < 0 \quad (8)$$

$$\forall i \in 1, \dots, N-2.$$

■

The  $N-2$  equations defined in the recurrence (8), together with the  $N-1$  equations given in (2) allow to obtain a nonlinear determined system of size  $2N-3$ , in which the unknowns are  $\{p_i\}_{i=2,\dots,N} \cup \{s_i\}_{i=1,\dots,N-1} - \{s_{\pi(1)}\}$ . As a consequence, it is possible to evaluate the performance (continuity and latency) for each particular permutation.

Observe that the identity permutation and the inverse one define Rarest First and Greedy strategies respectively, and we have inside each family of  $(N-1)!$  strategies the classical ones. Furthermore, a simple permutation shows that the mixture is also included in this family. Choose:

$$\pi(i) = i, i = 1, \dots, m; \quad (9)$$

$$\pi(i) = N - (i - m), i = m + 1, \dots, N - 1. \quad (10)$$

Moreover, it is possible to approximate the ascendent-descendent behavior of a given strategy function, as it is explained next.

**Property IV.2.** “*Approximation Strategy Property*”:

For each injective sequence  $s_i, i = 1, \dots, N-1$  there exists an element of permutation strategies  $s_{per}$  such that  $\forall i, j : s_i > s_j$  then  $s_{per(i)} > s_{per(j)}$ .

*Proof:* Be  $s$  any injective sequence. Then there exists a permutation of indexes  $i_k$  such that  $s_{i_1} > s_{i_2} > \dots > s_{i_{N-1}}$ . Let us call  $s_{per}$  to the strategy with permutation  $i_k$ . Then, by the Property IV.1 it is obtained that:

$$s_{per(i_1)} > s_{per(i_2)} > \dots > s_{per(i_{N-1})}.$$

The thesis is evident by observing the last two expressions. ■

Constructing the permutation strategy in the previous way assures that the increasing-decreasing behavior of every limited sequence may be simulated by a correspondent permutation strategy. Note that the injectivity is not a restrictive condition. By Property IV.1 all permutation strategies are injective. Moreover, we do not know any other deterministic feasible strategy.

The experience shows that an exhaustive study of all permutation strategies can be done only for limited buffer size, because of the prohibitive computational effort. Next we define a measure of optimality and a pseudodistance between permutations. This concepts together with the previous properties of our family, guide our strategy design, and permit to introduce a computationally efficient algorithm.

**Definition IV.3.** The quotient quality  $Q$  of a given monotonous probability of occupation  $p$  is defined as

$$Q = \frac{p_N}{L} = \frac{p_N}{\sum_{i=1}^N p_i}.$$

Observe that the quotient quality  $Q$  is monotonous increasing with respect to the continuity  $p_N$  and monotonous decreasing with latency  $L$ . The optimum permutation is then the one whose correspondent probability of occupation has major value of  $Q$ .

**Definition IV.4.** The pseudodistance between permutations is the minimum number of swaps needed to take one permutation in the other.

Based on these definitions, it is possible to create a systematic procedure for improving the performance. It receives a probability of occupation  $p$  and a number of iterations  $n$ , and returns a permutation  $\pi$ .

Figure 3 presents the pseudo-code of our procedure. This algorithm is oriented in results. This means that it is possible to input a desired probability of occupation  $p$ , and by approximating this input via an intermediate feasible probability of occupation, can improve the performance of this last. More specifically, through Steps 1, 2 and 3 (using the Approximation Strategy Property), it is obtained an intermediate permutation which achieves the nearest probability of occupation  $p^*$ . Immediately, the quotient quality of  $p^*$  is calculated (Step 4) and a greedy-flavored heuristic of local search is applied, in order to improve this last quality (Step 5). At this step we look in each iteration for the best neighbor permutation of the actual one, and then update. This process finishes when a number of iterations is reached, or when we cannot improve in the next step (referring that this permutation is better than any of its neighbors). It is worth to observe that in each iteration of the local search we need to measure  $C_2^{N-1} = \frac{(N-1)(N-2)}{2}$  qualities of permutations, versus a total of  $(N-1)!$  in a linear search between all permutations. The former is computationally treatable, and the algorithm gives a tradeoff between quality and time consumed.

```

Procedure PermutationFinderAlgorithm
Input: Probability of occupation:  $p_i, i = 1, \dots, N$ 
        Number of iterations:  $n$ 
Output: Permutation  $\pi$ 

{Construction of an initial permutation}
1: Obtain the ideal strategy sequence  $s_{ideal}(i)$  that achieves the latter
occupation probabilities  $p$ , through:

$$s_{ideal}(i) = \frac{p_{i+1} - p_i}{(1 - p_i)p_i}, i = 1, \dots, N - 1.$$


2: Find the permutation  $\pi$  that better approximates  $s_{ideal}$  such as
it was built in the Approximation Strategy Property IV.2.
3: Find solving the nonlinear system defined by equations (8) and (2),
the correspondent sequence of the permutation  $\pi$ , named  $s_{per}$ ,
and its sequence  $p^*$ .
4: Calculate the quality of  $p^*$ , defined by the quotient:

$$Q = \frac{p_N^*}{\sum_{i=1}^N p_i^*}$$


{Local Search cycle}
5: for each  $j \in \{1, \dots, n\}$  do
    Look for the best permutation  $\pi_{new}$  at distance 1 from  $\pi$ ,
    with quality  $Q_{new}$ .
    if  $Q_{new} > Q$  then
        update:  $Q \leftarrow Q_{new}$  and  $\pi \leftarrow \pi_{new}$ 
    else {this permutation is better than any neighbor}
        break and finish
    end if.
end for

```

Fig. 3. Permutation Finder Algorithm. A systematic procedure for improving the performance of the strategy. It receives a probability of occupation  $p$  and a number of iterations  $n$ , and returns a permutation  $\pi$ .

Now we focus in the initial seed of this algorithm. This is closely related with the concept of optimality previously defined, and an intuitive approximation of achievable probabilities of occupation. Next an ideal probability of occupation guides our initial decision.

**Lemma IV.5.** *Be  $p^* : p_i^* = \frac{1}{M}, \forall i = 1, \dots, N - 1, p_N^* = 1$  the ideal probability of occupation. Then, it is the optimum monotonous sequence with respect to the quotient quality  $Q$ .*

*Proof:* In order to find the optimum for the quotient quality, observe that the objective function is monotonous decreasing with respect to  $p_i$ , for each index lower than  $N$ . As a consequence, they must be chosen as small as possible, complying with the constraints of (2), i.e.  $p_i = 1/M$ . By the other hand, the objective function  $Q(p)$  is monotonous increasing with respect to the continuity  $p_N$ , which implies that  $p_N$  must be the highest possible, i.e.  $p_N = 1$ . This solution results in the ideal probability of occupation,  $p^*$ . ■

**Lemma IV.6.** *The ideal probability of occupation,  $p^*$ , is not realizable.*

*Proof:* For every probability of occupation, its correspondent strategy must obey equation (2). In particular, evaluating

at  $i = N - 1$ , the associated strategy must comply that:

$$s_{N-1} = \frac{p_N^* - p_{N-1}^*}{(1 - p_{N-1}^*)p_{N-1}^*} = \frac{1 - 1/M}{(1 - 1/M)1/M} = M > 1.$$

This last is impossible, since  $s_{N-1}$  is a probability. ■

Then, a desired initial vector should have high value of  $p_N$  and a slow value in the other indexes. As long as  $p^*$  is not realizable, a hurried solution would be to enter an exponential curve, because of its regularity and slow values in the beginning of the buffer. However, if  $p_i = M^{-\frac{N-i}{N-1}}$  we get using equation (2) that:

$$\log_M(\frac{s_{i+1}}{s_i}) = \log_M(\frac{1 - p_i}{1 - p_{i+1}}) > \log_M(1) = 0 \quad (11)$$

then the correspondent strategy is monotonous increasing, and the permutation built in the proof of the Approximation Strategy Property IV.2 is exactly  $\pi(i) = N - i$ , falling in the Greedy strategy. Experimentally, the local search of Step 5 shows that Greedy is better than any of its neighbors. But its poor continuity motivates us to find better results.

It is interesting to observe that the largest jump from  $p_i$  to  $p_{i+1}$  (in order to obtain high continuity) is possible when  $p_i \approx \frac{1}{2}$  and the peak of the piece selection function could be chosen in this index:

$$p_{i+1} - p_i = (1 - p_i)p_i s_i \leq \frac{s_i}{4} < \frac{1}{4}.$$

We model this desired jump of the probability of occupation as a change in slopes. The design is more flexible with the generation of two peaks. Moreover, in this way it is possible to approximate the behavior of the exponential curve, not falling in the known Greedy strategy. The proposed seed for the algorithm with its output can be seen in Figure 4. It can be noticed that the output sacrifices continuity in order to reduce latency. A comparison between the classical strategies and the new one proposed is illustrated in Figure 5 and Table I for the case of buffer size  $N = 20$  and  $M = 100$  peers. It can be appreciated that while the probability of occupation keeps as high as Rarest First, its latency is substantially lower.

The two outstanding properties of the algorithm proposed is that it is oriented in results, and its output permutation is better than the one which approximates the input as best as possible.

TABLE I  
PERFORMANCE OF DIFFERENT STRATEGIES. IT CAN BE APPRECIATED  
THAT OUR PERMUTATION FINDER ALGORITHM KEEPS THE PROBABILITY  
OF OCCUPATIONS HIGH AS RAREST FIRST, WHILE ITS LATENCY IS  
SUBSTANTIALLY LOWER.

Strategies	Continuity	Latency	Quotient
Rarest First	0.9251	11.5449	0.0801
Greedy	0.8157	3.0309	0.2691
Permutations	0.9223	4.7535	0.1940

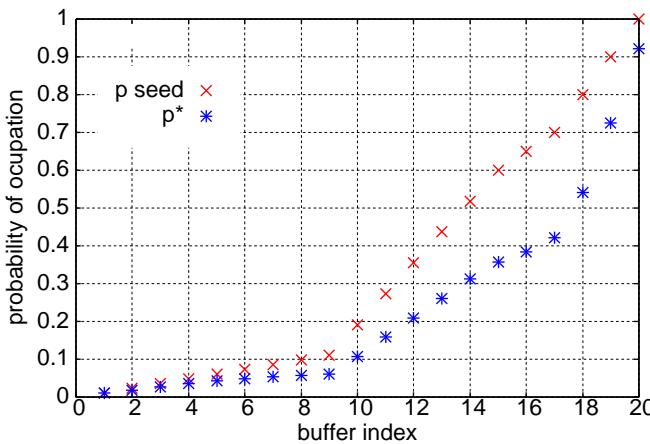


Fig. 4. Initial seed of the algorithm and its output. It can be noticed that the output sacrifices continuity in order to reduce latency

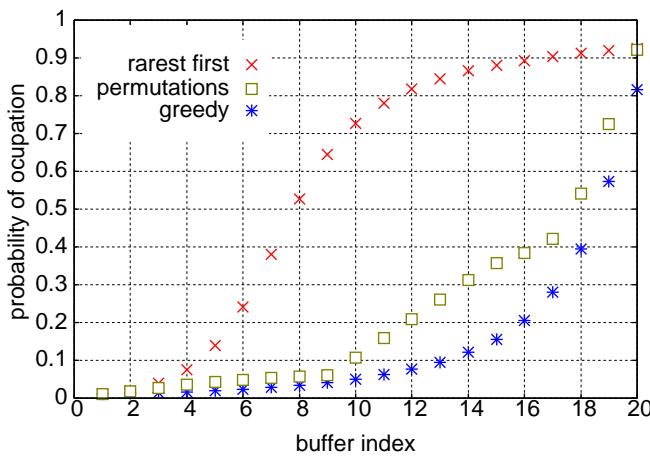


Fig. 5. Comparison between different strategies.

## V. CONCLUSIONS

In this paper a pull-process for cooperation was studied mathematically in depth. The main contributions are the new family of piece selection strategies based on permutations of the linear order of piece request, and the Permutation Finder Algorithm, which provides a systematic procedure for improving performance. This Algorithm exploits the strength of the Approximation Strategy Property, which gives a constructive way of approximating the shape of every injective strategy. Finally, the Permutation Finder Algorithm was applied to a particular input. It achieved similar continuity and lower latency than Rarest First.

## ACKNOWLEDGMENTS

This work was partially supported by project “Sistema eficiente de distribución de video y TV en tiempo real”, for the national telephony operator ANTEL.

## REFERENCES

- [1] C. D. Cranor, M. Green, C. Kalmanek, D. Shur, S. Sibal, J. E. V. der Merwe, and C. J. Sreenan, “Enhanced streaming services in a content distribution network,” *IEEE Internet Computing*, vol. 5, no. 4, pp. 66–75, 2001.
- [2] S. Wee, W. Tan, J. Apostolopoulos, and S. Roy, “System design and architecture of a mobile streaming media content delivery network (mscdn),” Streaming Media Systems Group, HP-Labs, Tech. Rep., 2003.
- [3] P. Rodríguez-Bocca, “Redes de Contenido: Taxonomía y Modelos de evaluación y diseño de los mecanismos de descubrimiento de contenido.” Master’s thesis, Universidad de la República, Facultad de Ingeniería, Intituto de Computación. ISSN 0797-6410 INCO-RT-05-13, Montevideo, Uruguay, October 2005.
- [4] S. Horovitz and D. Dolev, “LiteLoad: Content unaware routing for localizing P2P protocols,” in *Proceeding of the IEEE International Symposium on Parallel and Distributed Processing (IPDPS’08)*, Miami, USA, April 2008, pp. 1–8.
- [5] D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu, “Peer-to-peer computing,” HP Labs, Tech. Rep. HPL-2002-57, 2002. [Online]. Available: [citeseer.ist.psu.edu/milojicic02peertopeer.html](http://citeseer.ist.psu.edu/milojicic02peertopeer.html)
- [6] GoalBit Home page, <http://goalbit.sf.net/>, 2008.
- [7] M. E. Bertinat, D. D. Vera, D. Padula, F. Robledo, P. Rodríguez-Bocca, P. Romero, and G. Rubino, “Goalbit: The first free and open source peer-to-peer streaming network,” in *LANC ’09: Proceedings of the 5th international IFIP/ACM Latin American conference on Networking*. New York, USA: ACM, 2009.
- [8] S. Saroiu, P. K. Gummadi, and S. D. Gribble, “A measurement study of peer-to-peer file sharing systems,” in *Multimedia Computing and Networking*, 2002.
- [9] D. Zeinalipour-Yazti and T. Folias, “A quantitative analysis of the gnutella network traffic,” Department of Computer Science, University of California, Riverside, Tech. Rep., 2002.
- [10] K. Sripanidkulchai, B. Maggs, and H. Zhang, “An analysis of live streaming workloads on the internet,” in *IMC ’04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM Press, 2004, pp. 41–54.
- [11] A. Sentinelli, G. Marfia, M. Gerla, L. Kleinrock, and S. Tewari, “Will iptv ride the peer-to-peer stream?” *Communications Magazine, IEEE*, vol. 45, pp. 86–92, 2007.
- [12] S. Tewari and L. Kleinrock, “Analytical model for bittorrent-based live video streaming,” in *4th IEEE Consumer Communications and Networking Conference (CCNC’07)*, Las Vegas, NV, January 2007, pp. 976–980.
- [13] S. Alstrup and T. Rauhe, “Introducing octoshape - a new technology for large-scale streaming over the internet,” European Broadcasting Union (EBU), Tech. Rep., 2005.
- [14] P. Rodríguez-Bocca, “Quality-centric design of Peer-to-Peer systems for live-video broadcasting,” Ph.D. dissertation, INRIA/IRISA, Université de Rennes I, Rennes, France, april 2008.
- [15] S. Mohamed and G. Rubino, “A study of real-time packet video quality using random neural networks,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 12, pp. 1071–1083, December 2002.
- [16] B. Cohen, “Incentives build robustness in bittorrent,” [www.bramcohen.com](http://www.bramcohen.com), vol. 1, pp. 1–5, May 2003.
- [17] Y. Zhou, D. M. Chiu, and J. Lui, “A Simple Model for Analyzing P2P Streaming Protocols,” in *Proceeding of the IEEE International Conference on Network Protocols (ICNP’07)*, Beijing, China, October 2007, pp. 226–235.
- [18] A. Legout, G. Urvoy-Keller, and P. Michiardi, “Rarest first and choke algorithms are enough,” in *IMC ’06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2006, pp. 203–216.
- [19] C.-J. Wu, C.-Y. Li, and J.-M. Ho, “Improving the download time of bittorrent-like systems,” in *IEEE International Conference on Communications 2007 (ICC 2007)*, Glasgow, Scotland, June 2007, pp. 1125–1129.