A Simple Proactive Provider Participation Technique in a Mesh-Based Peer-to-Peer Streaming Service

María Elisa Bertinat, Darío Padula, Franco Robledo Amoza, Pablo Rodríguez-Bocca, and Pablo Romero

Laboratorio de Probabilidad y Estadística, Facultad de Ingeniería.
Universidad de la República. Montevideo, Uruguay.
dpadula@fing.edu.uy,
WWW home page: www.lpe.edu.uy

Abstract. The design of efficient protocols for mesh-based Peer-to-Peer (P2P) networks has many challenges, one of them is bandwidth allocation. On one hand, peers demand high Quality of Experience and network traffic when they watch their streaming contents. On the other, Internet Service Providers (ISPs) support their business with the capacity of their international links. A recent strategy considered in order to meet both requirements is the Proactive Provider Participation, shortly named P4P [1], [2]. This paper addresses the multi-objective P4P problem for live video content. First, we introduce a Ford-Fulkerson-based solution, that solves the P4P problem when only one content is distributed. Second, a greedy randomized technique is applied when multiple contents are shared. Finally, our algorithm is tested in a real multiple-content scenario, showing that the results highly outperform current strategies.

Keywords: Peer-to-Peer, P4P, Network Optimization.

1 Introduction

An important amount of today's Internet traffic is due to P2P networks, particularly for live video streaming [3,4]. For this reason, several peer-to-peer streaming networks were developed in the last years. The most successful ones are PPlive, TVUnetwork, SopCast, with proprietary and unpublished mesh-based protocols [3]. These are virtual networks developed at the application layer, over the Internet infrastructure. The nodes, called *peers*, offer their resources to others, basically because they share interests in common. The users can connect and disconnect freely. This makes P2P networks an attractive tool for them, but increases P2P's design challenges, because the bandwidth resource fluctuates uncontrolled. In mesh-based protocols, the cooperation is the key element in order to assure a certain quality of experience to end-users [3]. There are three main steps in all mesh-based P2P protocols for cooperation. First, when a peer enters the net it should discover other peers sharing the same content, which is called *swarm selection strategy*. Once a new peer knows other peers in his swarm, he must select the best ones to cooperate, what is called *peer selection strategy*. Finally, it should decide which pieces of the streaming content should be asked first, called the *piece selection strategy* [5], [6]. This paper is focused on the swarm selection strategy and in

the peer selection strategy. The main issue is to allocate the largest amount of traffic in the network without bottlenecks, and keeping the quality of experience between peers. In Sect. 2 the mathematical P4P model is explained (based on [1]). The reader can find related work on P4P in [7], [8], [9]. Section 3 contains a polytime resolution for one content. A Greedy Randomized heuristic [10] is proposed in Sect. 4 for the general case. In Sect. 5, we introduce this algorithm in a real P2P platform, called GoalBit [11].

2 Mathematical Model

This model is inspired in [1], where the authors simplify the problem into a linear programming one in order to solve it.In order to represent the complexity and scale of a real scenario with millions of peers, the peers are grouped in nodes. Each node is a geographical subset of Internet (for example: an autonomous system or an ISP point-of-presence), and they are interconnected by real links. Inside each node could be several peers sharing contents. Consider a network $N = (V, C)$ with nodes set $V = \{v_1, \ldots, v_n\}$, and a non-negative matrix with null diagonal $C = (c_{i,j})$ such that for each pair of different nodes v_i and v_j there are two one-way links, with capacities $c_{i,j}$ and $c_{j,i}$. The upload and download bandwidths are u_i^k and $d_i^k, i = 1, \ldots, n$ respectively, where $k \in \{1, \ldots, K\}$ represents different contents (each node v_i has K possible contents to download). Each link (i, j) uses a certain percentage of its capacity due to other applications, which is denoted by $b_{i,j}$ (called the background traffic). Be $\mathcal{P} = \mathcal{P}_1^{k_1}, \ldots, \mathcal{P}_m^{k_m}$ a set of oriented paths in the network, where $\mathcal{P}_h^{k_h} = (x_h, \ldots, y_h)$ (x_h is the uploader and y_h is the downloader). Be $t_1, \ldots, t_m$ their respective traffic magnitudes. In words: x_h uploads a traffic magnitude t_h of content type k_h to y_h by the oriented path $\mathcal{P}_h^{k_h}$. We assume that the intermediate nodes does not consume bandwidth. The ISPs objective (1) is to reduce the bottleneck over the most expensive edges in Internet, subject to the satisfaction of peers in the network (first constraint). The other constraints express that the set of oriented paths must be feasible (not exceeding the link's capacities neither the bandwidths). We will assume a practical *intra-domain traffic* hypothesis: $u_i^k = 0$ or $d_i^k = 0$ for each node v_i. Basically, the network works in steady state, where all the traffic uploaded is downloaded [1], [2].

$$\min_{\mathcal{P}} \max_{(i,j):i\neq j} \rho(\mathcal{P}) = b_{i,j} + \frac{\sum_{h:(i,j)\in\mathcal{P}_h^{k_h}} t_h}{c_{i,j}}, \qquad s.t. \tag{1}$$

P4P Model

$$\begin{cases} \max_{\mathcal{P}} \sum_{h=1}^{m} t_h & (2) \\ \sum_{h:x_h=i,k_h=k} t_h \leq u_i^k, \forall i \in V, k \in \{1, \ldots, K\} & (3) \\ \sum_{h:y_h=j,k_h=k} t_h \leq d_j^k, \forall j \in V, k \in \{1, \ldots, K\} & (4) \\ b_{i,j}c_{i,j} + \sum_{h:(i,j)\in\mathcal{P}_h}^{k_h} t_h \leq c_{i,j}, \forall i \neq j \in V & (5) \end{cases}$$

In practice, once we have a set of oriented paths $\mathcal{P}$ and their respective magnitudes, it is possible to converge probabilistically to that traffic distribution in a real network. In order to reach this goal the following strategies can be followed: the swarm for a peer located at node v_i, that asks for content k, is populated with the following proportion of peers from node v_j: $w_{ji}^k = \frac{\sum_{h:(j,i)\in\mathcal{P}_h^k} t_h}{\sum_{x\in V}\sum_{h:(x,i)\in\mathcal{P}_h^k} t_h}$. Moreover, the peer statistically takes in consideration these weights also in his peer selection strategy in order to have a faster converge. See [12] for details.

3 A Polytime Resolution for One Content

Although the high complexity of the general P4P formulation, we show there is a Fully Polynomial Time Approximation Scheme (FPTAS) when one content is distributed in the network (i.e. $K = 1$ in the P4P Model) in Sect. 3. These algorithm is naturally extended with a greedy randomized technique discussed in Sect. 4.

Definition 1. *An algorithm is a Fully Polynomial Time Approximation Scheme (FPTAS) for the optimization problem P if for any given instance I and $\epsilon > 0$, it finds in polynomial time in the input size I and in $\frac{1}{\epsilon}$ a solution S, which complies that $|val(S) - val(I)| < \epsilon \times val(I)$, where $val(I)$ is the optimal value for instance I.*

Algorithm 1 $\mathcal{P} = MaxFlow((V, C), B, u, d, \epsilon)$

```
1: C^net = C × (1 − B)
2: N_in = (V, C^net)
3: N_aux = Extend(s, t, N_in, u, d)
4: (P_M, φ_max) = FordFulkerson(N_aux)
5: ρ_min = max_{i≠j} b_ij
6: ρ_max = 1
7: while |ρ_max − ρ_min| > ε do
8:    ρ = (ρ_min + ρ_max)/2
9:    C^net = C × (ρ × 1 − B)
10:   Update(N_aux)
11:   (P, φ) = FordFulkerson(N_aux)
12:   if φ = φ_max then
13:      ρ_max = ρ
14:   else
15:      ρ_min = ρ
16:   end if
17: end while
18: return P
```

Algorithm $MaxFlow$ constructs an auxiliary network connecting every node of the original network $N = (V, C)$ with two ideal nodes s and t, and finds the minimum link utilization of the original network preserving at the same time the maximum flow (using

Ford-Fulkerson Algorithm iteratively). It receives a network $N = (V, C)$, two vectors u and d that represent the upload and download bandwidths for each node, the background matrix B and a given tolerance $\epsilon > 0$ from Definition 1. Lines $1 - 2$ substract the background traffic keeping the remainder C_{net} and define a network $N_{in} = (V, C_{net})$. In Line 3 an auxiliary network N_{aux} is constructed by extending N_{in} with Function $Extend$. It connects two ideal nodes s (the *source*) and t (the *terminal*) to N_{in}, where s (resp. t) is connected with every node $v_i \in V$ with its upload (download) capacity u_i (resp. d_i). Immediately in Line 4, a max-flow ϕ_{max} is found in N_{aux} between s and t, calling the classical Ford-Fulkerson Algorithm, where $\mathcal{P}$ is the set of output paths and their flow magnitudes (recall that Ford Fulkerson returns the min-cut, but a maximal flow can be saved as well). Next, the main idea is to find the minimum percentage link utilization ρ that achieves ϕ_{max}. The block of Lines $5 - 17$ consists of a bipartition search in the closed interval $[\rho_{min}, \rho_{max}]$, where $\rho_{max} = 1$ (all resources are used) and $\rho_{min} = max_{i \neq j} b_{ij}$ (no resources are used), until the desired tolerance ϵ is met. In each iteration, the capacities C^{net} of the auxiliary network N_{aux} are modified, according with ρ. Note that the logic of the bipartition preserves the maximum flow ϕ_{max}.

Theorem 1. *Algorithm $MaxFlow$ is a FPTAS for the P4P Problem when $K = 1$*

Proof. Consider the set of output paths (with magnitudes) $\mathcal{P}$ in the auxiliary network N_{aux} and remove the ideal nodes s and t. The proof consists of three steps: feasibility, optimality and computational effort.

1. We assert that *$\mathcal{P}$ is a feasible set of paths for the P4P model*: In effect, it is clear that ϕ_{max} is preserved during the entire Algorithm $MaxFlow$, and by construction it is the maximum flow in the original network, satisfying constraint 2. Constraints 3 and 4 are fulfilled since the links capacities between s, t and every node v_i in the original network are chosen with respective capacities u_i and d_i. Constraint 5 is satisfied: set $\rho : \rho_{min} \leq \rho \leq 1$. The total flow for link (i, j) does not exceed its capacity, since Line 9 assures that $c^{net}_{i,j} = c_{i,j} \times (\rho - b_{ij})$ and:

$$b_{i,j}c_{i,j} + \sum_{h:(i,j)\in\mathcal{P}_h}^{k_h} t_h \leq b_{i,j}c_{i,j} + c^{net}_{i,j} \leq c_{i,j}\rho \leq c_{i,j},$$

   where $b_{i,j}c_{i,j}$ is the total traffic inside link (i, j) given to other applications.
2. Observe that *$\mathcal{P}$ is an optimal set of paths*. The Ford-Fulkerson algorithm finds the maximum flow between the nodes s and t in the auxiliary network N_{in}, which has a direct interpretation in the original network N.
3. Finally, *$MaxFlow$ is a polytime algorithm*: since Ford-Fulkerson is polynomial in the number of nodes n, it suffices to show that the number of iterations i in the bi-partition search is polytime in n and $1/\epsilon$. Given $\epsilon > 0$, there exists an integer i such that $2^i > \frac{1}{\epsilon}$. Consequently, $MaxFlow$ runs in polytime with n and in $i < \lfloor log_2 1/\epsilon \rfloor < 1 + \frac{1}{\epsilon}$ steps, which is a linear expression in $\frac{1}{\epsilon}$ as required in Definition 1. These inequalities hold for any given $\epsilon > 0$, so the global error can be bounded as well as the relative error.

In conclusion, $MaxFlow$ is a FPTAS for the P4P problem when one content is delivered, as we wanted to prove.

4 A Metaheuristic Approach for Multiple Contents

To the best of our knowledge, there is not an exact resolution in polytime for multiple contents [12]. A possible metaheuristic approach is detailed next.

Algorithm 2 $\mathcal{P}_{RP} = RandPriority(U, D, C, B, p)$

1: $\mathcal{P}_{RP} = \varnothing$
2: **while** $(length(p) > 0$ AND $C \geq 0)$ **do**
3: k = $ChooseContent(p)$
4: $\mathcal{P}_{RP} = \mathcal{P}_{RP} \cup MaxFlow((V, C), B, U^k, D^k)$
5: $Update(U, D, C, B, p)$
6: **end while**
7: **return** $\mathcal{P}_{RP}$

Algorithm $RandPriority$ is very simple, and proposes a Greedy Randomized generalization for multiple contents [10]. It receives the bandwidth matrices U and D (that store the bandwidth of every node for each content), the capacity and background matrices C and B respectively and a probability vector p_k that measures the priority to the content type k. In Line 1, the set of paths $\mathcal{P}_{RP}$ is empty. In each iteration, a content is chosen randomly according with the priority vector p (Line 3). Immediately, in Line 4 Algorithm $MaxFlow$ is called, in order to find the best routing for that content. The obtained flows are added to $\mathcal{P}_{RP}$, and the bandwidth and capacities updated in Line 5 (the entry k in the priority vector p is deleted, so content k is not considered any more). The process is repeated until there is no more capacity or after all contents were delivered. In order to get a reference of the performance achieved by $RandPriority$, it is desired to find an upper bound for the maximum traffic. We can easily translate every multiple-content instance into a songle-content one, identifying all contents as the same, and considering new upload and download bandwidths $u_i^{'}$ and $d_i^{'}$ for each node v_i [12]:

$$u_i^{'} = \sum_k u_i^k; \quad d_i^{'} = \sum_k d_i^k$$

In this way, the total traffic in the new single-content network ϕ_B is certainly not lower than the original traffic ϕ for the corresponding multiple content instance.

Definition 2. *Given an instance for the general P4P Model, the performance coefficient for an algorithm that achieves link utilization ρ and total traffic ϕ is defined by:*

$$\alpha = \rho \frac{\phi_B}{\phi}, \tag{6}$$

where ϕ_B is the traffic in the single-content network (obtained identifying contents) found applying Algorithm $MaxFlow$.

Algorithm $BestPaths$ combines these ideas, calling $RandPriority$ iteratively, and returning the best output (considering the performance coefficient, which is measured in each iteration).

Algorithm 3 $\mathcal{P}_{BP} = BestPaths(U, D, B, C, p, Iter, \phi_B)$

1: **for** $i = 1$ to $Iter$ **do**
2: $(\mathcal{P}_i) = RandPriority(U, D, B, C, p)$
3: $(\rho_i, \phi_i) = Calculate(\mathcal{P}_i)$
4: $\alpha_i = \rho_i \frac{\phi_B}{\phi_i}$
5: **end for**
6: $\alpha_j = min_{1 \leq i \leq Iter} \alpha_i$
7: **return** $\mathcal{P}_{BP} = \mathcal{P}_j$

5 Empirical Results

5.1 Performance in a Real Platform for One Content

In this work we implemented Algorithm $MaxFlow$ to converge empirically to the optimum P4P routing in a real P2P video streaming platform, called GoalBit [11], when only one content is distributed. It is known that BitTorrent works in downloading, but does not comply real time requirements for streaming applications [6, 13]. GoalBit maintains the BitTorrent's philosophy, trying to extend its success to video streaming. The routing protocol we propose acts exactly in the moment of the peer list conformation, applying $MaxFlow$ algorithm to skew routing to converge to the theoretical P4P solution. Emulations were carried out with information provided by The Uruguayan National Telephony Operator ANTEL[1] from their GoalBit deployed live service. We contrast the peer selection strategy using Algorithm $MaxFlow$ versus the Classical GoalBit strategy (based on BitTorrent) [11]. For all emulations we evaluate the quality of experience of final users (buffering time), the total amount of exchanged traffic and the one which crosses international links. In particular, we show the results of two emulations for the cases of 60 and 100 simultaneous peers connected in average.

Tables 1 and 2 show, for both strategies, the total traffic (incoming and outgoing international links traffic[2]). These tables also show the percentage growth of incoming and outgoing traffic $P_{G_{in}}$ and $P_{G_{out}}$ when the P4P model is applied, in relation with a Classical routing. Specifically:

$$P_{G_{in}} = 100 \times (\frac{In_{P4P}}{In_{Classic}} - 1); \quad P_{G_{out}} = 100 \times (\frac{Out_{P4P}}{Out_{Classic}} - 1),$$

where In_{P4P}, $In_{Classic}$ and Out_{P4P} and $Out_{Classic}$ represent the total incoming and outgoing traffic for P4P and Classic models respectively. It is desirable to obtain negative percentage growth, interpreted as a reduction in the international links, and consequently, an improvement in relation with the Classical routing strategy.

[1] This paper was supported by The National Telephony Operator ANTEL (www.antel.com.uy)
[2] As we can only measure the traffic through the outgoing and incoming links to and from Uruguay, this will be our reference node.

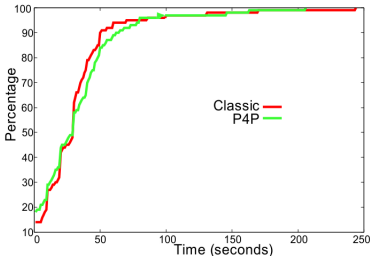

Fig. 1. Buffering time for 60 peers

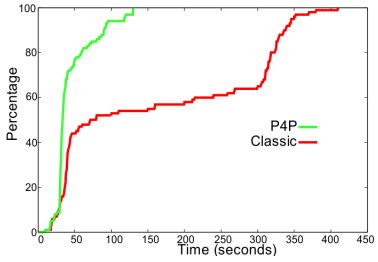

Fig. 2. Buffering time for 100 peers.

Table 1. Link utilization for 60 peers.

Model	*Incoming*	*Outgoing*	*Total*
Classic	31656	31366	183069
P4*P*	7927	16446	183067
$\%growtraffic$	-47.57	-74.96	0.0

Table 2. Link utilization for 100 peers.

Model	*Incoming*	*Outgoing*	*Total*
Classic	5681	10253	55078
P4*P*	3657	4451	58893
$\%growtraffic$	-56.59	-35.63	6.93

It can be appreciated from Table 1 that the incoming reduction is 47.57%, while the outgoing reaches 74.96%, keeping the total traffic achieved by the Classical routing. Also, Table 2 shows important reductions for the 100 simultaneous peers case, and an increasing in total traffic is perceived. Figure 1 shows that the buffering time distributions are quite similar for both techniques. In both cases, the 85% of peers perceived a buffering time lower than 55 seconds. In contrast, for the 100 simultaneous peers case, P4P present much lower buffering times when compared with the Classical strategy (see Fig. 2). A 68% of peers wait no more than 38 seconds to start playing when P4P is applied, but only a 27% of peers can start playing the video during the same time. Many emulations were carried out for different inputs showing similar bandwidth savings, close to 30% in average [12].

5.2 Simulation for Multiple contents

In order to understand the performance of Algorithm $BestPaths$, we developed a simple simulation, taking uniformly random bandwidth matrices $U = n \times Rand(n,k)$, $D = n \times r \times Rand(n,k)$, background and capacity matrices $B = 0$ and $C = k \times s \times Rand(n,n)$. The integers n and k are the number of nodes and contents in the network respectively; the factors $r > 1$ and $s > 1$ regulate how pessimistic or optimistic the scenario is. Note that an increment in these factors mean higher expected download bandwidths and capacities. We called $BestPaths$ taking $Iter = 10$ in order to find better results.

In Table 3 and Table 4 we specified (by the parameters n, k, r and s) a set of scenarios. Also, these tables show the results of the simulation. We ran 5 times each scenario in order to avoid random effects, and count the number of times the bound is reached (column $\%Success$ of Tables 3 and 4). They also show for each scenario

the mean and standard deviation for: output link utilization ρ, (μ_ρ, σ_ρ), output traffic ϕ (μ_ϕ, σ_ϕ), and quotient between ϕ and the respective upper bound ϕ_B (called μ_{ϕ/ϕ_B} and σ_{ϕ/ϕ_B} respectively). It can be appreciated that the bound is reached in every optimistic instance (when $r = s = 2$). Interestingly, a success can be afforded in pessimistic scenarios also (see for example Row 3 of Table 4, when the resources were limited with $r = 1.2$ and $s = 1.5$, and the optimum was achieved in all instances).

Table 3. $BestPaths$ vs $Bound$ for different instances when $r = 2$ and $s = 2$

$Scenario$	n	k	r	s	μ_ρ	σ_ρ	μ_ϕ	σ_ϕ	$\mu_{\frac{\phi}{\phi_B}}$	$\sigma_{\frac{\phi}{\phi_B}}$	$\%Success$
1	20	5	2	2	0.9169	0.0959	1012.8	67.56	1.0000	0.0000	100
2	20	10	2	2	0.8022	0.0411	2040.4	54.16	1.0000	0.0000	100
3	40	5	2	2	0.8716	0.0807	4042.4	106.95	1.0000	0.0000	100
4	40	10	2	2	0.7361	0.0854	8068.3	278.12	1.0000	0.0000	100

Table 4. $BestPaths$ vs $Bound$ for different instances when $r = 1.2$ and $s = 1.5$.

$Scenario$	n	k	r	s	μ_ρ	σ_ρ	μ_ϕ	σ_ϕ	$\mu_{\frac{\phi}{\phi_B}}$	$\sigma_{\frac{\phi}{\phi_B}}$	$\%Success$
1	20	5	1.2	1.5	0.9907	0.0207	1008.7	31.83	0.9961	0.0043	20
2	20	10	1.2	1.5	0.9651	0.0482	1943.0	105.90	0.9978	0.0028	40
3	40	5	1.2	1.5	1.0000	0.0000	4036.8	125.42	1.0000	0.0000	100
4	40	10	1.2	1.5	0.9500	0.0314	7935.9	212.31	0.9952	0.0057	20

6 Conclusions

In this work, the Proactive Provider Participation (P4P) model was analyzed. When all peers are interested in exactly one content, the problem can be solved with the desired accuracy, combining a Ford-Fulkerson approach and a bi-partition scheme. However, a greedy randomized technique was developed for the general case, inspired in the success for single content distribution. It selects contents randomly and allocates bandwidth according with the previously methodology for one content. Emulations contrasting a bandwidth allocation in a real platform (GoalBit) and the P4P resolution are very challenging. They indicate a 30% link utilization reduction in average with our P4P application, and at the same time the quality of experience seems to improve. We carried out simulations for optimistic and pessimistic scenarios when multiple contents are delivered. An upper bound for the maximum traffic was introduced, identifying all contents as the same. Every optimistic instance achieves 100% success, meaning that the maximum flow is sent at a minimum link utilization running the greedy randomized algorithm for multiple contents. This ideal upper bound is also achieved in every pessimistic scenario, at least once. Qualitatively, this highlights the competitiveness of the P4P model for bandwidth allocation.

References

1. Xie, H., Krishnamurthy, A., Silberschatz, A., Yang, Y.R.: P4P: Proactive Provider Participation for P2P. Technical report, Yale University, Department of Computer Science (2007)
2. Xie, H., Yang, R., Krishnamurthy, A., Liu, Y., Silberschatz, A.: P4P: Provider Portal for Applications. In: Conference on Data Communication, SIGCOMM.
3. Rodríguez-Bocca, P.: Quality-centric design of Peer-to-Peer systems for live-video broadcasting. PhD thesis, INRIA/IRISA, Université de Rennes I, France (april 2008)
4. Cisco Systems, Inc.: Hyperconnectivity and the Approaching Zettabyte Era. (June 2010)
5. Bertinat, M.E., Vera, D.D., Padula, D., Robledo, F., Rodríguez-Bocca, P., Romero, P., Rubino, G.: A COP for Cooperation in a P2P Streaming Protocol. In: Proceedings of the IEEE International Conference on Ultra Modern Telecommunications(ICUMT'09), Washington, DC, USA, IEEE Computer Society (12-14 October 2009) 1–7
6. Romero, P.: Optimización de la Estrategia de Selección de Piezas de Video en Redes P2P. Master's thesis, Facultad de Ingeniería, Montevideo, Uruguay (November 2009)
7. Yang, G., Sun, Y., Wu, H., Li, J., Liu, N., Dutkiewicz, E.: A Trust Model in P4P-integrated P2P Networks based on domain management. In: Computers and Communications (ISCC), IEEE Symposium. (june 2010)
8. Guo, Z., Min, L., Yang, H., Yang, S.: An Enhanced P4P-Based Pastry Routing Algorithm for P2P Network. (august 2010)
9. Guo, Z., Yang, H., Yang, S.: P4P Pastry: A novel P4P-based Pastry routing algorithm in peer to peer network. In: Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference. (april 2010)
10. Resende, M.G.C., Ribeiro, C.C.: Greedy Randomized Adaptive Search Procedures. In F. Glover and G. Kochenberger, editors, Handbook of Methaheuristics, Kluwer Academic Publishers (2003)
11. Bertinat, M.E., Vera, D.D., Padula, D., Robledo, F., Rodríguez-Bocca, P., Romero, P., Rubino, G.: Goalbit: The first free and open source peer-to-peer streaming network. In: Proceedings of the 5th international IFIP/ACM Latin American conference on Networking, New York, USA, ACM (september 2009) 49–59
12. Padula, D.: Compromiso entre Pares e ISPs en el contexto P4P: Optimización en dos niveles. Master's thesis, Facultad de Ingeniería, Montevideo, Uruguay (July 2010)
13. Dale, C., Liu, J.: A measurement study of piece population in bittorrent. In: IEEE Global Telecommunications Conference, GLOBECOM, New York, USA, IEEE (2007) 405–410