A Framework for Highly Reconfigurable P2P Trackers

Pedro Sousa

Original scientific paper

***Abstract*—The increasing use of Peer to Peer (P2P) applications, usually ruled by selfish behaviors, is posing new challenges to the research community. As contributions of this work we firstly devise a general framework underpinning the development of highly reconfigurable P2P trackers. Following that, a novel tracker architecture is proposed and several illustrative and enhanced tracker configurations are described. As result, the devised solution turns possible that flexible, programmable and adaptive peer selection mechanisms can be introduced at the P2P application level.**

The proposed solution assumes the general framework of one of the most popular P2P solutions, in this case a BitTorrent-like approach. As illustrative examples of the proposed framework capabilities, several straightforward and easy to deploy tracker configuration examples are presented, including methods for qualitative differentiation of swarm peers and advanced P2P Traffic Engineering mechanisms fostering the collaboration efforts between ISPs and P2P applications. Both the framework and the devised tracker configurations are validated resorting to simulation experiments.

***Index Terms*—Communications Software, Programmable Trackers, Context-aware P2P Applications, Traffic Engineering, Network Optimization**

I. Introduction

P2P overlay networks [1] can be considered as distributed systems where peers form self-organized network infrastructures built on top of the physical network topology. The massive use of P2P protocols, along with the use of distinct peering strategies, have dramatically changed the traffic profile, also introducing new problems for ISPs (Internet Service Providers) and posing new research challenges. As illustrative example, several studies show that protocols such as BitTorrent [2] has evolved into one of the most popular networks and is now responsible for more than one third of all Internet traffic, thus being an excellent case study in the P2P area [3], [4].

However, P2P applications may cause traffic to scatter and some connections may unnecessarily traverse multiple links of the provider network. This may lead to higher load in some network links also having several negative implications from the ISP point of view. Moreover, unnecessary inter-domain traffic might be generated, causing that some non-tier-1 providers be forced to relay a substantial volume of traffic between its providers, leading to possible disruptions of ISPs economics [5], [6]. To reinforce this argument, some studies of commonly used applications (e.g. Skype) confirmed such problems [7], [8]. Also, several Traffic Engineering approaches used by ISPs need to estimate traffic matrices in order to achieve efficient routing configurations. This estimation effort could be made difficult due to the variability of the P2P dynamics, making complex the demand matrix estimation [9], [10]. In order to face such problems, several techniques are used by ISPs. The location of caching devices to reduce bandwidth consumption by the peers is just one example [11], [12]. However, due to the large number of P2P protocols, such approaches are protocol dependent and may not be considered as general solutions for the P2P versus ISP coexistence problem. Another example is the use of traffic control mechanisms to relieve the influence and variability of the P2P traffic inside the network. Once again, this may not fulfill both the ISP and users objectives, as P2P performance will be degraded and users expectations will not be satisfied. Moreover, some of these techniques require meticulous packet level inspection to detect P2P traffic adding complexity to the network [13].

In this context, this work proposes a collaborative framework exploring the concept of highly programmable P2P trackers, illustrating its behavior within the context of a BitTorrent-like system. In such systems the tracker is responsible for coordinating the file distribution, namely by informing to which peers a recently arrived peer should connect to download the pieces of the file. The devised framework aims to be used by highly reconfigurable applications and services based on the P2P paradigm. The objective of the proposed tracker entity is to assist several application level configuration mechanisms, namely the peer selection tasks. With the proposed solution it will be possible to foster the development of flexible peer selection mechanisms [14], also attaining enhanced service quality differentiation at the P2P application level and enriching P2P systems with collaborative Traffic Engineering solutions involving both applications and ISPs entities. Such flexible P2P configurations, sustained by very simple and easy to deploy tracker programming schemes, clearly contribute for widening and enhancing the results of other P2P optimization architectures (e.g. [5], [15], [16]). As result, the proposed framework is a powerful solution allowing administrators to better control, regulate and differentiate P2P traffic dynamics within the network domain. Although this proposal focuses on BitTorrent like systems, other P2P software solutions may also readapt the ideas and mechanisms here discussed.

The paper is organized as follows: Section II presents the proposed framework rationale with the concept of programmable P2P trackers. Section III provides examples of distinct tracker configurations easy to implement in real sce-

Manuscript received May 24, 2013; revised January 20, 2013.

Pedro Sousa is with Centro Algoritmi and Department of Informatics, University of Minho, Braga, Portugal (email: pns@di.uminho.pt).

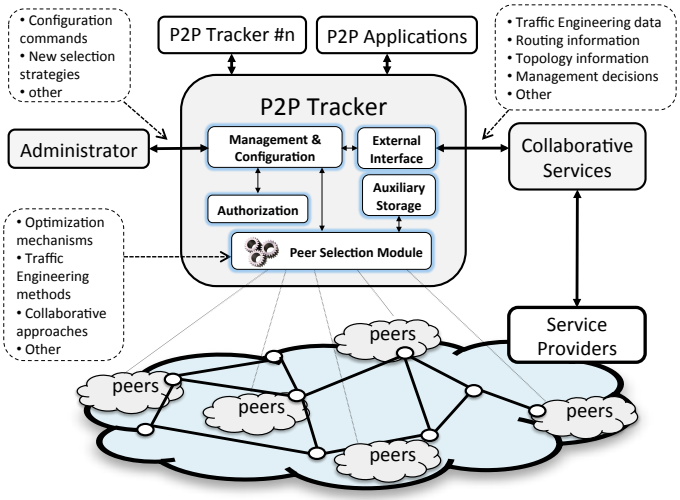

Fig. 1. Devised framework for programmable P2P trackers.

narios. Section IV presents a set of simulation experiments with the corresponding results. Finally, Section V concludes the presented work.

II. Programmable P2P Tracker Architecture and Functionalities

The proposed concept of context aware and programmable trackers is presented in the framework depicted in Figure 1, also highlighting the internal structure of the P2P tracker. The framework considers the general case of applications following a BitTorrent-like approach and the use of a P2P tracker to rule the swarm behavior (swarm is the adopted term to identify a set of peers connected to a given torrent). Nowadays, such system principles can be also used to develop proprietary applications used by content/service providers to interact with their end-users.

In the example of Figure 1 a single centralized tracker is presented, however, the use of multiple trackers is possible in order to improve the system scalability, given that appropriate coordination and state information is exchanged among the trackers. The tracker is able to be configured by the administrator, or other authorized external services, and to interact with collaborating external services providing valuable cross-layer information or other service-specific data. In the devised framework, the envisaged application scenario assumes that the tracker is the only entity able to provide peering information, i.e. client side software provided to end-users is not able to exchange peer identities with other peers, thus the tracker fully controls the peering informations provided to the clients[1]. As reward of this end-user limitation, ISPs are expected to provide users with incentives such as give preferential traffic treatment to such conforming P2P applications, in counterpoint with more aggressive techniques (e.g. inspection, blocking, shaping) usually adopted to deal with undesirable P2P traffic.

[1]Some classical P2P systems can also optionally behave in this way, disabling any sort of external peer exchange (e.g. with PEX or DHT) for a given torrent. As example, in Azureus/Vuze this can be done by tracker admins using the key *private* in the *.torrent* file.

In BitTorrent-like classical systems, new peers wishing to join a specific swarm contact a tracker which provides the clients with a random sample of peers (usually a refresh timer is also defined at the client side to control subsequent requests). This sample is used by the peers for establishing new P2P connections with other peers in the swarm to download a given shared resource. After this stage, several BitTorrent rules will drive the data transfer processes among the peers. These additional details about the BitTorrent protocol regarding pieces selection algorithms and choking strategies to determine which peers to choke/unchoke can be found in [1], [2], [17]. The P2P BitTorrent tracker main role is then to keep track of the current peers participating in a given swarm and dynamically provide random peer samples to newly arrived peers in the swarm, thus having a crucial influence in the behavior of this type of P2P applications. In the proposed framework, the several modules integrating the P2P tracker (in Figure 1) are:

Peer Selection Strategies : This module holds a repository with alternative peer selection mechanisms able to be used by the tracker. These peer selection strategies have been previously defined by the administrator (or other external entity) and were programmed, uploaded and activated using appropriate configuration commands. Each one of these mechanisms are possible to be activated and (re)configured during the application lifetime and may require distinct types of interactions with external information sources, depending on their specific objectives.

Tracker Management/Configuration: This module acts as the interface for all the configuration procedures. Thus, it receives commands responsible for tasks such as: selecting the active peer selection strategy; uploading alternative peer selection strategies; defining which external entities should be contacted by the tracker to gather additional information; etc. The configuration procedures can be performed by an administrator or follow an automated approach, where applications or external entities allowed to interact with the tracker provide on-the-fly (re)configuration commands. Moreover, tracker configurations may also be static or dynamic, with the later allowing the change of the peer selection strategy during the swarms lifetime.

External Interface: The module responsible for controlling the tracker interactions with all the external information sources interacting with the P2P system. This modules behaves as an intermediary between the tracker and the external entities, thus supporting specific protocols required to communicate with such entities. Several examples of possible external information sources are: network level entities/services able to provide privileged network level information; Provider Portals (e.g. as defined in [15]); general information related with established agreements with other networks or service/content providers requiring particular service quality differentiation at the P2P level; Traffic Engineering related information, among many other possibilities.

Auxiliary Storage: This module stores several auxiliary data characterizing active peers in a given swarm and other related state information, such as the data retrieved from external sources.

Authorization: Deals with authorization and security issues that are associated with the tracker procedures. This may include: validate the identity of the users/entities issuing configuration commands to the tracker; certifying the tracker iterations with authorized external entities; validating the peer requests, among other related procedures.

In a summarized perspective, several scenarios could take advantage of the capabilities of the proposed framework: *i) Collaborative networking approaches:* with the objective of developing collaborative efforts among several entities (e.g., ISPs and P2P level applications) it will be possible the development of context aware P2P solutions, reacting and modifying the application behavior in accordance with specific inputs (e.g. traffic loads, time periods, service level agreements, etc.); *ii) P2P service quality differentiation:* the development of strategies for the qualitative differentiation of the service levels provided to different peers, allowing that several policies could be defined by ISPs, given incentives or penalizing specific peers in a given swarm. Such differentiation capabilities also open new opportunities allowing that ISPs motivate end users to adopt collaborative behaviors; *iii) Traffic engineering mechanisms:* the proposal also benefits the optimization efforts related with traffic engineering tasks. Given the flexible and programmable nature of the P2P tracker it will be possible to configure P2P applications in order to better estimate and plan how P2P traffic will affect the underlying network.

III. Tracker Configuration Examples

This section describes some illustrative tracker configurations. The presented mechanisms, which can be easily implemented by software developers in real scenarios, are based on simple tracker configuration logics. The presented examples include service qualitative differentiation, location aware mechanisms, advanced Traffic Engineering approaches and mixed configurations (see summary in Table I).

A. Service Differentiation - penalizing peers in a swarm

This tracker configuration example illustrates a peer selection behavior that might be used as a pure penalizing mechanism able to punish non-conforming peers with some pre-defined P2P application level rules or, due to specific agreements with other entities (e.g. ISPs, content providers), punish peers which behavior is degrading the overall performance of the network. However, other scenarios may also benefit from these differentiation capabilities, such as: the need of controlling the traffic generated by a given set of peers; the need of forcing P2P connections only among a specific set of peers; the ability to provide distinct service levels for content distribution scenarios, among others. There are several possibilities to be adopted to penalize (or influence) peering decisions resorting to specific tracker configurations. As example, it is possible to configure the tracker in order to restrict the number of peers in the samples returned to specific clients (in this case low priority peers). In BitTorrent-like applications, during data exchange processes, peers have a maximum number of simultaneously open connections with other peers. Thus, a simple approach would be to return a peer sample with a small number of peering possibilities, hindering that the maximum number of open connections can be reached by such low priority peers. As consequence, low priority peers will have a reduced opportunity of discovering and connecting to other peers in the swarm and, comparatively with peer samples having a higher dimension, such swarm elements are expected to experience lower service quality levels. Additionally, the tracker should retain a timestamp of the last request made by such low priority peers and decide to provide, or not, new reduced samples during the subsequent requests. An illustrative pseudo-code of the tracker configuration under this operation mode is presented in Algorithm 1. The magnitude of such restriction limits, both in the peer sample dimensioning ($peer\_limit$ input of the function $reduced\_swarm\_peer\_sample()$ in line 4) and allowed renewal sample period ($time\_limit$ in line 3), will influence the penalization degree imposed to low priority peers. In order to prevent service starvation these strict restrictions affecting low priority peers might be gradually relieved by the tracker during the swarm lifetime.

B. Service Differentiation - benefiting peers in a swarm

Another possible tracker configuration allows to provide incentives to specific peers in a given swarm through a careful selection of the peers included in the samples returned by the tracker. This incentive based selection mechanism can be used with several objectives in mind, such as: allow that preferential treatment is given to a specific set of peers; divert traffic from specific crucial links or paths of the network by carefully placing seeds in strategic points of the network; the definition of enhanced high priority sub-swarms where a restricted set of peers has access to high upload capacity seeds, among others possibilities. Such positive discrimination mechanisms could be used for pure quality differentiation purposes, but also with side-effects traffic engineering purposes, where both content and network providers exchange valuable informations to reach an attractive peering configurations.

Algorithm 2 presents an illustrative pseudo-code of this tracker configuration. In the presented rationale the tracker is able to define high priority sub-swarms integrating a given set of peers. Such peers are selected by the tracker according with a given predefined criteria, and they will take advantage of specific conditions induced by the tracker to improve their service quality levels. In this example, the tracker manipulates the peer samples that are retuned to such high priority peers and has the ability to introduce new incentives for that high priority group (in line 4 the $decision\_rule$ input of $add\_additional\_incentives()$ function will guide such decision during the swarm lifetime). A simple example of a possible incentive is the inclusion of high upload capacity seeds that are only visible for a given group of privileged peers. Moreover, non-priority peers will receive peer samples that should not include peers/seeds integrating high priority sub-swarms, as in that case they will also indirectly take advantage of already distributed data pieces among high priority peers (the function $exclude.high.priority.peers()$ in line 7).

TABLE I
ILLUSTRATIVE TRACKER CONFIGURATIONS MECHANISMS

Mechanism Type	Implemented Method	Description	Experiments
Service Differentiation	*Penalizing peers in a swarm()*	Section III-A	Section IV-A
	Benefiting peers in a swarm()	Section III-B	Section IV-B
Location Aware	*Decreasing inter-domain traffic()*	Section III-C	Section IV-C
Advanced Traffic Engineering	*P2P Link Impact Estimation()*	Section III-D1	Section IV-D1
	P2P Link Protection()	Section III-D2	Section IV-D2
Hybrid Mechanisms	*Hybrid Configuration()*	Section III-E	Section IV-E

Algorithm 1 Service Differentiation - Penalizing peers: *get_peer_sample(peer p, swarm s)*

```
pri ← get_peer_priority_group(p, swarm)
if pri == low_priority_group then
    if first_request(p, s) or (current_timer() - last_request_timer(p, s)) ≥ time_limit) then
        peer_sample ← reduced_swarm_peer_sample(s, peer_limit)
    else
        peer_sample ← null
    end if
    last_request_timer(p, s) ← current_timer()
else
    peer_sample ← random_swarm_peer_sample(s)
end if
update_swarm_info(p, s, pri)
return(peer_sample)
```

Algorithm 2 Service Differentiation - Benefiting peers: *get_peer_sample(peer p, swarm s)*

```
pri ← get_peer_priority_group(p, swarm)
if pri == high_priority_group then
    peer_sample ← priority_swarm_peer_sample(s, pri)
    peer_sample ← add_additional_incentives(peer_sample, pri, decision_rule)
else
    peer_sample ← random_swarm_peer_sample(s)
    peer_sample ← exclude.high.priority.peers(peer_sample, pri)
end if
update_swarm_info(p, s, pri)
return(peer_sample)
```

C. Location aware optimization - decreasing inter-domain traffic

This particular configuration mode is mainly devised for a simple Traffic Engineering purpose, with the P2P tracker trying to reduce the inter-domain traffic generated by a given swarm. In this case the tracker was programmed to gather information about the location of current peers in a specific swarm along with the location of newly arrived peers requesting the tracker services. Such information may be provided by network level entities cooperating with the P2P level. When receiving a request from a new peer, the tracker was programmed to return a random sample of peers in the swarm, but now taking into account two distinct operational phases. First, if the swarm is in an initial state (or with a limited number of peers) then the default behavior is assumed, i.e. the return of a random sample of the existing peers. The current number of peers in the swarm or other P2P level information stored at the tracker might be used to assess the state of the swarm. Otherwise, if the swarm is not considered to be in an initial state then the returned sample will be mainly composed by peers belonging to the same networking domain of the requesting peer. This strategy intends to drastically reduce the inter-domain traffic generated by P2P applications without noticeable degradation of the service quality. The first phase of this mechanism allows that diverse peering relations occur independently of peers locations, assuring a minimum level of data pieces distribution. From that point on, newly arrived peers will mainly use local peers to download the distributed resource.

An illustrative pseudo-code of this configuration is presented in Algorithm 3. Here, the tracker will resort to external network level entities that will provided location related in-

Algorithm 3 Location aware - Avoiding inter-domain traffic: *get_peer_sample(peer p, swarm s)*

```
1: l ← get_peer_location(p)
2: if swarm_in_initial_state(s) then
3:    peer_sample ← random_swarm_peer_sample(s)
4: else
5:    peer_sample ← build_location_aware_sample(s, l, threshold)
6: end if
7: update_swarm_info(p, s)
8: return(peer_sample)
```

formation of the peers. Based on that, after an initial phase of the swarm lifetime, the tracker will return peer samples based on the location of the requesting swarm peer in order to minimize inter-domain traffic exchanges. For that purpose the $build\_location\_aware\_sample()$ function (in line 5) will take into account the location of the requesting peer (l), the corresponding swarm identification (s) and a threshold value ($threshold$) that is used to tune the mixture nature of the return sample, i.e. the amount of peers from other domains that are allowed to be included in the peer sample.

D. Advanced Traffic Engineering Configurations

This section presents other Traffic Engineering configurations taking advantage of the interoperation capabilities of the proposed architecture. Lets assume, as illustrative example, that it is intended that the tracker should be capable of facing the following challenges: *i)* for a given swarm composition (or subset of peers) forecast which are the network links that will be more influenced by P2P traffic. This scenario might be useful to raise the tracker with the capability of estimating the network level consequences of the combination of peering adjacencies resulting from the random samples returned by the tracker (see section III-D1); *ii)* allow tracker configurations that protect specific network links from excessive levels of P2P traffic (see section III-D2). In order to accomplish such objectives, there some metrics from the graph theory field (e.g. [18], [19]) that, if correctly adapted, can constitute valuable inputs to face such challenges.

1) P2P Link Impact Estimation: In this configuration mode, the tracker gathers topological and routing information from collaborating services and represents ISP routers (N) and transmission links (L) in a simple graph $G = (N, L)$. For simplicity, network scenarios with symmetric links are assumed, thus being represented by an undirected graph. Each pair of nodes ($x, y \in N$) is connected by a path, according to the routing strategy adopted in the network (e.g. shortest-path based). Each link ($l \in L$) has routing weights which are used by the ISP to compute shortest-paths among the nodes. The location area of end-users peers participating in the swarm is identified by the location of the corresponding ISP access router, a, with $a \in A$ and $A \subseteq N$. For the estimation of the P2P link impact values, the tracker evaluates a betweenness centrality measure for each one of the links, considering the locations (area) of the swarm peers, identified by the corresponding ISP router. For a specific link, l, and a pair of end-users areas, $i, j \in A$, the metric considers the ratio between the number of shortest paths from i to j, $nsp_{i,j}$, and the number of such paths that pass through link l, $nsp_{i,j}(l)$. Each link l is then assigned with a partial impact value of $\frac{nsp_{i,j}(l)}{nsp_{i,j}}$ for the case of i, j peering adjacencies. Summing all the partial impact values involving link l, a reference value within the interval $[0, 1]$ is obtained, considering all the possible area peering adjacencies combinations, i.e. $|A| \cdot (|A| - 1)$. In the case of P2P swarms where end-user areas have a considerable unbalanced distribution of number of peers (also reflected in the number of peers from each area included in the random samples returned by the tracker) an additional weighting factor is introduced, $w_{i,j}$, for each specific i, j[2] case, increasing the importance of shortest paths connecting areas having a higher number of peers. As result, links with higher betweenness centrality values have a higher probability of being traversed by traffic of the considered BitTorrent P2P swarm. For the case of a tracker returning random samples to the contacting peers, Equation 1 presents the devised normalized P2P betweenness centrality value for link l, within the interval $[0, 1]$.

$$\frac{\sum_{i,j \in A, i \neq j} \frac{nsp_{i,j}(l)}{nsp_{i,j}} \cdot w_{i,j}}{|A| \cdot (|A| - 1)}, l \in L \quad (1)$$

The above mentioned metric can be further enhanced by considering that due to the inherent characteristics of the TCP protocol, BitTorrent peers usually have a considerable probability of establishing peering connections with nearest peers in the network, taking advantage of lower RTTs. In Equation 1, when considering a given shortest path between areas i and j (assuming the context of peers in area i trying get data from peers in area j), it is possible to assign a preference value[3] ($p_{i \leftarrow j} \in [0, 1]$ with $\sum_{j \in A, j \neq i} p_{i \leftarrow j} = 1$) to such shortest paths, implicitly expressing how close are areas j and i. This value is then multiplied by the number of possible external peering adjacencies that could be made by peers in a area, i.e. $|A| - 1$. The resulting value is used as a weighting factor when accounting shortest paths between areas i and j. Equation 2 expresses a more refined P2P betweenness

[2] $w_{i,j}$ factor considers the ratio between the number of peers involved in the area peering adjacency i, j over the total number of peers involved in all possible area peering adjacencies. In order to preserve the original form of the betweenness measure, this ratio is multiplied by $|A| \cdot (|A| - 1)$ for normalization purposes.

[3] If required, in highly heterogeneous scenarios, the estimation model could be further enriched by also reflecting in this parameter the relative quality of the average upload capacities of area j peers, when compared with other peers in the domain, along with other network condition related information.

Algorithm 4 P2P Link Impact Estimation: *get_link_impact_estimation(swarm s)*

1: $routing\_topology\_info \leftarrow$ get_routing_and_topology_info( )
2: $loc\_info \leftarrow$ get_area_routers_and_peers_location(s)
3: **for** (all network inks, l_z) **do**
4: $P2P\_IM(l_z) \leftarrow$ p2p_link_impact_estimation(loc_info, $routing\_topology\_info$, s, $p_{i \leftarrow j} values$, l_z) /* Comment: Based on Eq. (2) */
5: **end for**
6: return($P2P\_IM(l_z)$ values)

centrality value for link l, from this point on designated as P2P link Impact Measure (P2P_IM). If required, this measure could be announced to network services or administrators which may require the tracker to change its behavior according to a given objective. Algorithm 4 summarizes a simple tracker internal programming logic to achieve the above mentioned P2P link impact estimation.

$$\frac{\sum_{i,j \in A, i \neq j} [(|A|-1) \cdot p_{i \leftarrow j}] \cdot \frac{nsp_{i,j}(l)}{nsp_{i,j}} \cdot w_{i,j}}{|A| \cdot (|A|-1)}, l \in L \quad (2)$$

2) P2P Link Protection: The above mentioned P2P link impact metric can be also used by the tracker to protect specific network links from excessive P2P traffic. In that way, administrators or authorized services can inform the tracker about network links that they intend to protect, inducing the tracker to decrease the corresponding P2P impact values. Thus, for a specific set of protected links, $K \subseteq L$, the tracker should minimize the P2P impact values of links $k \in K$, i.e. forcing peering adjacencies constrained by the objective function in Equation 3.

$$\min \Big(\sum_{k \in K, K \subseteq L} P2P\_IM(k) \Big) \quad (3)$$

The underpinning optimization concept is that the P2P tracker be able to induce peering adjacencies that should now avoid traversing network paths including the protected links. It is possible that under some peering configurations achieved by the tracker the previously presented P2P link impact equations need to be adapted in consonance with the new conditions[4].

E. Hybrid Tracker configurations

All of the previously described configurations can be combined, resulting that the P2P tracker is able to behave in a kind of hybrid configuration mode, involving both service qualitative differentiation and intelligent peering mechanisms from the Traffic Engineering field. In addition, other novel tracker configuration could also be formulated. In this context, optimization (e.g. [20], [21], [22], [23]) or forecasting (e.g. [24]) mechanisms from the field of computational intelligence, which had been previously used to resolve other traffic engineering problems, are examples of possible tools over which additional tracker configurations could be formulated.

[4]As an example, if peers in some area are not able to contact peers in other specific areas, then the number of all possible area adjacencies will be no longer $|A| \cdot (|A|-1)$ as assumed in Equations 2, for normalization purposes.

IV. Experiments and Results

The ns-2 simulator [25] was used to implement a prototype of the proposed framework (Figure 1) and test the devised tracker configuration mechanisms (Section III). A packet-level simulation patch [26] implementing a BitTorrent-like protocol was used to give ground for the development of the proposed solution. The patch was extended in order to integrate a prototype of the programmable tracker architecture along with the illustrative peer selection strategies of Table I. Additional state information storage to assist peer selection decisions along with interfaces to simulate the interactions with external entities were also developed. Several debugging and log functionalities were also integrated in the tracker for simulation results analysis.

Figure 2 a) illustrates one network topology used to present some illustrative results of the proposed framework. At the top level the network is divided in three distinct areas interconnected by inter-area links. Each area is then composed by a second level of nodes/links which configurations allow the definition of each area internal structure. All the network parameters (e.g. inter-area/intra-area link capacities, upload/download access links capacities, propagation delays, etc.) may be configured at the simulation level. In Figure 2 a) the concept of an area may have two distinct interpretations. When testing selection mechanisms having the objective of reducing the inter-domain traffic an area will be assumed in fact as a networking domain, where links $D1 \rightarrow D2, D1 \rightarrow D3$ and $D2 \rightarrow D3$ will be viewed as interconnections between distinct domains. Otherwise, for simulations with tracker configurations disregarding domain related issues, the three areas will be interpreted as integrating an unique domain. In such cases, intra and inter area links will be viewed in fact as internal links of a domain, and their distinct capacities and propagation delays will be used to increase the heterogeneity of the domain topology. Most of the parameters controlling the BitTorrent-like protocol may be also configured, including parameters such as: the number of seeds and leechers per domain and their arrival processes into the swarm group, tracker related configurations, the use (or not) of superseeding, chunk size, file size, several timers guiding the P2P protocol, among others.

In the selected examples the results were taken from a simulation scenario assuming nearly 100 leechers per area,

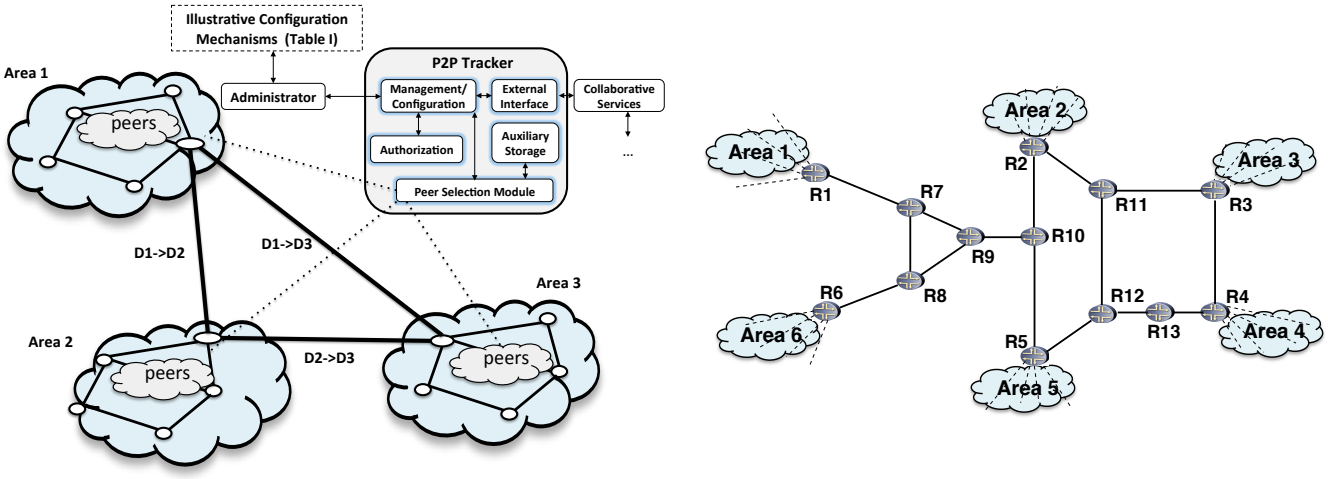

Fig. 2. a) Tracker architecture and network topology used in the experiments; b) A more complex network level topology for testing some tracker advanced Traffic Engineering configurations.

resulting in a total number of 300 peers. The file size is 50 MB and the chunk size 256 KB. The maximum number of peer addresses requested from the tracker is 25 (i.e. the peer sample size returned by the tracker to contacting peers), however depending on the selected mechanism the tracker may manipulate this value. Most of the selected results assume the worst case scenario for file dissemination, i.e. initially only one seed for a considerable number of leechers in the network (i.e. the flash crowd effect). Whenever possible super-seeding mode was used in the simulations. At the network level the peers have, on average, an upload capacity of 1 Mbps and a download capacity which is considered to be eight timer higher than this value (i.e in this case to simulate a common scenario with asymmetric access links, such as in ADSL or Cable access for home users). In order to improve the heterogeneity of each area, the propagation delays of the access links were randomly generated in the interval of 1-50 ms. In this specific scenario, the inter-area links were considered to be able to support a share of 10 Mbps for P2P traffic and their propagations delays are at least two times higher than the maximum value considered for intra-area links. In most of the presented results, the peers performance is measured taking into account the download time needed for a complete file transfer. In order to simplify the results visualization along the next sections, each peer is assigned with a $peer_{id}$ identification, in this case in the interval $(1, 300)$.

The scenario of Figure 2 b) shows a distinct network topology that was used specifically to test the advanced Traffic Engineering mechanisms (i.e. P2P link impact estimation and protection), involving a larger number of ISP core links, thus adding more complexity to the tested mechanisms. The next sections presents simulation results of the tracker configurations summarized in Table I. For each tracker configuration, five simulations were made, being the corresponding mean values presented.

A. Penalizing Peers in a Swarm

The results presented in Figure 3 a) and b) show two distinct scenarios where the tracker was programmed to penalized specific peers in a given swarm. This is done in the context of the discussed in Section III-A, using the configuration defined in Algorithm 1. Such mechanism restricts the number of peers included in the samples for penalized peers, also holding an auxiliary timestamp information at the tracker to progressively relieve such constraints during the swarm lifetime. In this specific case, for penalized peers, the number of peers returned in the samples halves the maximum number of simultaneously active connections for downloading data from other peers. By this way, these low priority peers experience a lower service quality level as they are constrained in the way they are able to establish P2P connections to get all the pieces of the original shared file. In the case of the example presented in Figure 3 a) several peers in all the networking areas were penalized. In this scenario, to make easier the results visualization, the tracker assumes that penalized peers are those having a $peer_{id}$ which is multiple of 5. In the second scenario, with the results plotted in Figure 3 b), specific peer groups within each network area were selected to be penalized ($peers_{id}$ in the intervals $(50, 75)$, $(150, 175)$ and $(250, 275)$ were considered as low priority). As observed, in both scenarios the tracker induced service qualitative differentiation, with low priority peers achieving higher download times.

B. Benefiting Peers in a Swarm

This section presents simulation results regarding the strategies discussed in Section III-B for benefiting specific P2P peers, and which mechanism was described in Algorithm 2. In such context, Figure 4 a) shows the results obtained using a programmable tracker configured to benefit two groups of peers, in this case belonging to the $peers_{id}$ intervals $(125, 135)$ and $(175, 185)$. In this case, the strategy adopted by the tracker is to include in the returned samples two high upload capacity seeds that are unaccessible to other peers in the swarm. Thus,

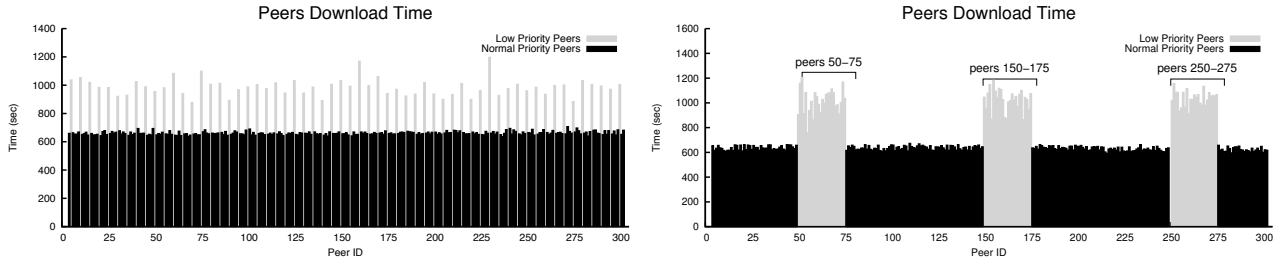

Fig. 3. Penalized peers with a) $peer_{ids}$ multiple of 5 and b) $peer_{ids}$ in the intervals $(50, 75)$, $(150, 175)$ and $(250, 275)$.

the tracker assures that peers in the mentioned intervals and such special seeds form a kind of high priority sub-swarm, i.e. they exchange data apart from the other peers of the swarm. As consequence, as plotted by Figure 4 a), differentiation of service qualitative levels is effectively achieved and high priority peers effectively obtain a better service quality at the P2P level (i.e. lower download times).

Results from a distinct configuration of the tracker operating under this mode are provided in Figure 4 b). In this example several levels of service differentiation are achieved. As observed in Figure 4 b) there are two classes of peers obtaining lower download times. The first group of peers ($Class_A$ priority peers with $peer_{ids}$ in $\{120, 140, 160, 180, 200\}$) receive the better overall service quality. Another distinct group ($Class_B$ priority peers with $peer_{ids}$ in $\{20, 40, 60, 80, 100, 220, 240, 260, 280, 300\}$) also obtain a positive discrimination, but with lower quality than $Class_A$. This multi-level positive differentiation was induced by the tracker by proving each one of the high priority sub-swarm with distinct quality levels of incentives (e.g. the number of hidden seeds, corresponding upload capacities, seed location advantages, etc.).

C. Location aware optimization - decreasing inter-domain traffic

In this example the tracker was programmed to behave in a collaborative perspective, receiving peer location information from external collaborative network level entities, with the objective of reducing the inter-domain traffic generated by the P2P level. For that purpose, the tracker was programmed with a similar configuration to the one illustrated in Algorithm 3 of section III-C. Figure 5 shows comparative swarm behavior results when the tracker is configured with the default selection mode and when is programmed with the collaborative location aware peer selection mechanism. As observed, when the tracker behaves within such intelligent peer selection strategy, the inter-domain traffic generated is at least twelve times lower than the observed in the default selection mode (see the three graphs plotting the generated inter-domain traffic in Figures 5 b) c) d) during the swarm lifetime). Moreover, and even taking into account that peer selection decisions are now more constrained and local peers have a higher probability of being selected from neighboring peers, the average download times of the peers were also improved (see Figure 5 a)), resulting by the the fact that BitTorrent data transfers use the TCP protocol, and connections having lower RTT values are expected to achieve higher throughput rates. This example illustrates the possibility of developing collaborative approaches effectively attending both P2P and ISPs objectives.

D. Advanced Traffic Engineering Configurations

This section presents illustrative simulation results obtained when the tracker is configured within the mechanisms described in Sections III-D1 and III-D2. As mentioned, the network topology used for these particular configuration modes is the one depicted in Figure 2 b).

1) P2P Link Impact Estimation: This specific example illustrates the tracker capabilities to estimate the quantitative P2P link impact values when configured with the mechanism described in Algorithm 4. The used network topology was depicted in Figure 2 b) and a single seed in considered to exist on each end-user network area. The values presented in Figure 6 a) presents the qualitative P2P link impact estimations devised by the tracker (within the interval $[0, 1]$) against the corresponding cumulative values of the traffic traversing each link after simulating a classical BitTorrent swarm behavior. As observed, the P2P traffic resulting from the swarm behavior has a major impact in some specific links of the network domain. The estimated P2P link impacts (using the $P2P\_IM$ metric[5] of Equtation 2) included in Figure 6 a) show a very acceptable match when their relative values are compared with the relative values among the measured traffic values. In fact, Figure 6 a) shows a similar trend among the link traffic values and the forecasted link impact values. In this perspective, and even considering that some distortions may exist in the link impact values when compared with measured traffic, external entities or administrators can rely on trackers that use the $P2P\_IM$ metric to nearly forecast the expected qualitative impact of P2P traffic in the network domain.

2) P2P Link Protection: The results included in this section illustrate a tracker configuration protecting specific links of the network from excessive P2P traffic, also using the network topology of Figure 2 b). In this scenario, it is assumed that the tracker was informed (e.g. by the network administrator) that it should protect the following links: $R7 \leftrightarrow R9$, $R8 \leftrightarrow R9$ and $R9 \leftrightarrow R10$. When configured within such constraints the tracker resorts to Equation 3 minimizing the impact values of the link set K, with $K = \{R7 \leftrightarrow R9, R8 \leftrightarrow R9, R9 \leftrightarrow R10\}$. During the optimization process the tracker will devised the

[5]In the experiments, the estimation model used $p_{i \leftarrow j}$ set to 0.4 for the nearest area and 0.15 for the other areas.

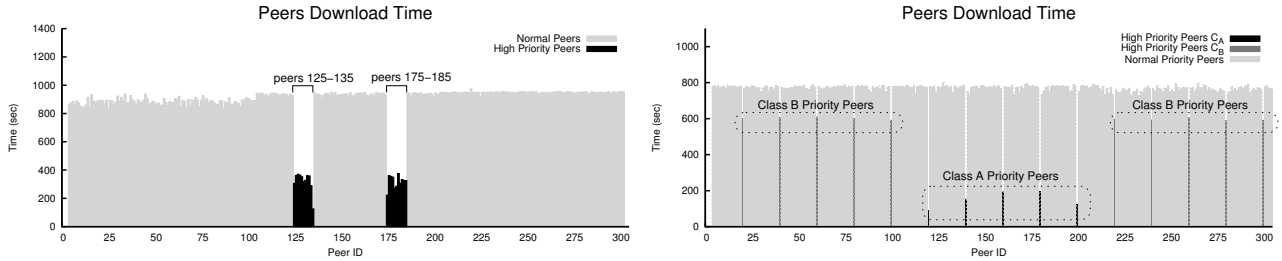

Fig. 4. a) Benefited peers are the ones with $peer_{ids}$ within the intervals $(125, 135)$ and $(175, 185)$; b) An example of multi-level positive differentiation of specific swarm peers.

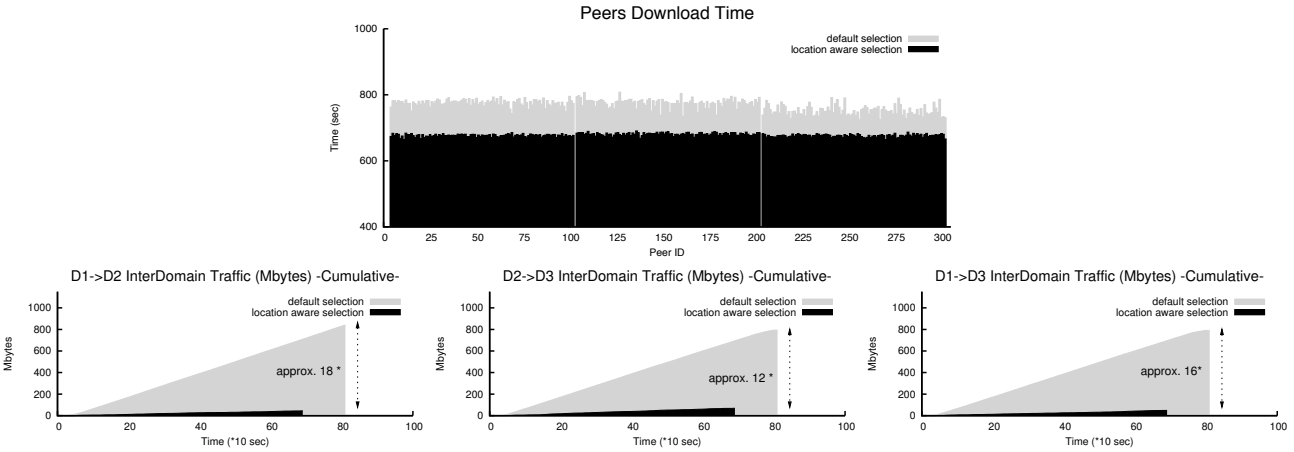

Fig. 5. Collaborative optimization: a) download times and b) c) d) inter-domain traffic in the links $D1{\rightarrow}D2$,$D1{\rightarrow}D3$,$D2{\rightarrow}D3$.

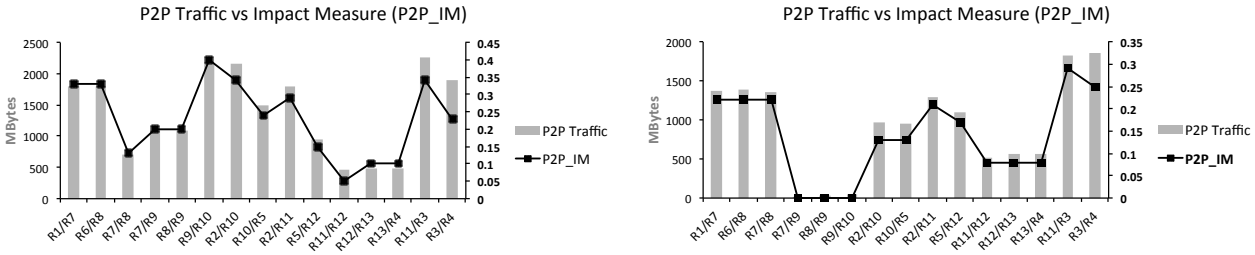

Fig. 6. P2P Traffic versus P2P Link Impact Estimation a) Default scenario b) Tracker configured to minimize P2P impact in links $R7 \leftrightarrow R9$, $R8 \leftrightarrow R9$ and $R9 \leftrightarrow R10$ (for network topology of Figure 2 b)).

most appropriate peering adjacencies that should be induced by the returned peer samples. In this specific case the tracker defines two independent peering groups one with peers from areas 1 and 6, and another one with peers from areas 2, 3, 4 and 5. This solution will completely avoid traffic from the P2P swarm to traverse the defined links, i.e. impact measures equal to zero, however, other not so severe solutions could also be attained by the tracker. Figure 6 b) shows the estimated P2P link impact values evaluated by the tracker after the optimization process triggered by the administrator. As observed, it is visible in the $P2P\_IM$ metric values that protected links will suffer no impact by the the P2P swarm. Again, as depicted in Figure 6 b), such qualitative impact estimations corroborate the cumulative P2P traffic values which traversed each link during the swarm simulation. Thus, links $R7 \leftrightarrow R9$, $R8 \leftrightarrow R9$ and $R9 \leftrightarrow R10$ were effectively protected from the P2P swarm behavior, only presenting almost imperceptible values of P2P traffic[6].

E. Hybrid Configurations

In the last selected example the tracker is configured in a hybrid configuration mode. For that purpose, the results of Figure 7 were obtained with the tracker programmed to benefit a specific group of peers in the network area two ($peers_{id}$ in the interval $(150, 160)$) and to penalize a group of peers in the network area one ($peers_{id}$ in the interval $(20, 30)$). In this way the tracker was configured with a mixed peer selection configuration comprising the aforementioned Algorithms 1 and 2. As observed in Figure 7, the results clearly show the

[6] These residual values are due to the implemented algorithm at the tracker, with an initial phase where no constraints are applied to the peering adjacencies.

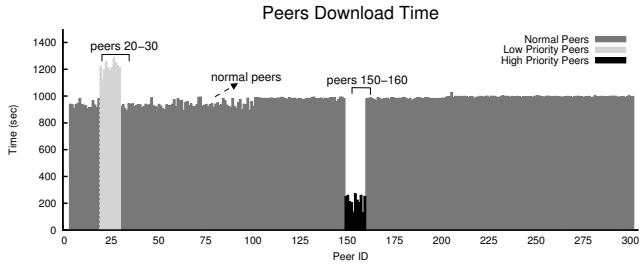

Fig. 7. Mixed configurations: penalized peers in the interval $(20, 30)$ and benefited peers in the interval $(150, 160)$.

correctness of the devised hybrid mode, showing that mixed service qualitative differentiation is possible to be achieved.

V. Conclusions

This work explored and proposed the concept of programmable trackers to raise P2P based Internet applications with extended capabilities, assuming for that purpose an application context based on a BitTorrent-like system. After describing the inherent functionalities of the proposed tracker architecture, some examples of advanced peer selection configurations were presented and illustrated resorting to simulation.

The presented simulation results highlighted the tracker capabilities in order to: *i)* sustain the development of P2P collaborative behaviors, namely in the effort of decreasing inter-domain traffic; *ii)* make possible to achieve P2P service qualitative differentiation, where a given set of peers will receive better service quality; *iii)* introduce advanced P2P Traffic Engineering models, allowing ISPs to estimate the link impact that P2P traffic will have on the network infrastructure and protect specific network links from P2P traffic. Furthermore, as also corroborated by simulation results, hybrid configurations are also supported by the tracker.

Given the flexibility and the wide range of the programmable alternatives that might rule the tracker behavior, either configured by the administrators or by accredited external entities, several improvements are now possible to be achieved. In particular, this proposal also clearly benefits the development of advanced P2P-based applications able to foster intelligent and collaborative efforts between the ISPs and the P2P applicational level.

Acknowledgments: This work is financed with FEDER funds by the Programa Operacional Fatores de Competitividade COMPETE and with national funds by FCT Fundação para a Ciência e Tecnologia for the project: FCOMP-01-0124-FEDER-022674.

References

[1] Lua, K., Crowcroft, J., Pias, M., Sharma, R., Lim, S.: A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys & Tutorials*, vol 7, Issue 2, pp. 72-93 (2005).

[2] Choen, B.: Incentives build robustness in BitTorrent. *In Proceedings 1st Workshop on Economics of Peer-to-Peer Systems*, Berkeley (Jun. 2003).

[3] Karagiannis, T., et al.: Is p2p dying or just hiding?. *In Proceedings of GLOBECOM*, Dallas, USA, (Nov. 2004).

[4] Schulze, H., Mochalski, K.: Internet Study 2007: The Impact of P2P File Sharing, Voice over IP, Skype, Joost, Instant Messaging, One-Click Hosting and Media Streaming such as YouTube on the Internet. *Technical Report* (2007).

[5] Xie, H., Krishnamurthy, A., Silberschatz, A., Yang, Y. R.: P4P: explicit communications for cooperative control between P2P and network providers, *http://www.dcia.info/documents/P4P_Overview.pdf* (2008).

[6] Seetharaman, S., Ammar, M.: Characterizing and mitigating inter-domain policy violations in overlay routes. *In Proceedings of IEEE International Conference on Network Protocols (ICNP)* (2006).

[7] Guha, S., Daswani, N., Jain, R.: An experimental study of the skype peer-to-peer VoIP system. *In Proceedings of IPTPS* (Feb. 2006).

[8] Xie, H., Yang, Y. R.: A measurement-based study of the skype peer-to-peer VoIP performance. *In Proceedings of IPTPS*, Bellevue (Feb. 2007).

[9] Keralapura, R., Taft, N., Chuah, C., Iannaccone, G.: Can ISPs take the heat from overlay networks?. *in Proceedings of HotNets-III*, San Diego, CA (Nov. 2004).

[10] Qiu, L., Yang, Y. R., Zhang, Y., Shenker, S.: SelFIsh routing in Internet-like environments. *In Proceedings of SIGCOMM*, Karlsruhe, Germany (Aug. 2003).

[11] Shen, G., Wang, Y, Xiong, Y., Zhao, B. Y., Zhang, Z.: HPTP: Relieving the tension between ISPs and P2P. *In Proceedings of IPTPS* (Feb. 2007).

[12] Wierzbicki, A., Leibowitz, N., Ripeanu, M., Wozniak, R.: Cache replacement policies revisited: The case of p2p trafc. *In Proceedings of GP2P*, Chicago, IL (Apr. 2004).

[13] Spognardi, A., Lucarelli, A., DiPietro, R.: A Methodology for P2P File-Sharing Trafc Detection. *In Proceedings of the Second International Workshop on Hot Topics in Peer-to-Peer Systems 2005 (HOT-P2P 2005)*, pp. 52- 61 (Jul. 2005).

[14] Sousa, P.: Flexible Peer Selection Mechanisms for Future Internet Applications. *In Proceedings of BROADNETS 2009 - Sixth International ICST Conference on Broadband Communications, Networks and Systems*, Madrid, Spain (2009).

[15] Xie, H. et al: P4P: Provider Portal for Applications. *In Proceedings of ACM SIGCOMM 2008*, August 17-22, Seattle, Washington, USA (2008).

[16] Choffnes, D. R., Bustamante, F. E.: Taming the Torrent: A practical approach to reducing cross-ISP traffic in P2P systems. *In Proceedings of the International ACM SIGCOMM conference* (Aug. 2008).

[17] Legout, A., et al: Clustering and Sharing Incentives in BitTorrent Systems. *In Proceedings of ACM SIGMETRICS'2007*, June 12-16, San Diego, USA (2007).

[18] Opsahl, T., Agneessens, F., Skvoretz, J.: Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, vol. 32, Number 3, pp. 245-251 (2010).

[19] Narayanan, S.: The betweenness centrality of biological networks. *MSc Thesis*, Faculty of the Virginia Polytechnic Institute and State University (2005).

[20] Rocha, M., Sousa, P., Cortez, P., Rio, M.: Quality of Service Constrained Routing Optimization using Evolutionary Computation. *Applied Soft Computing Journal*, Vol. 11, Issue 1, pp. 356-364, Elsevier (2011).

[21] Sousa, P., Rocha, M., Rio, M., Cortez, P.: Efficient OSPF Weight Allocation for Intra-domain QoS Optimization. In: Parr, G., Malone, D., O Foghlu, M. (eds.), IPOM 2006. LNCS, Vol. 4268, pp. 37-48. Springer, Heidelberg (2006)

[22] Sousa, P., Cortez, P., Rio, M., Rocha, M.: Traffic Engineering Approaches Using Multicriteria Optimization Techniques. *In Proceedings of WWIC 2011 - 9th International Conference on Wired/Wireless Internet Communications*, pp. 104-115, Springer, LNCS 6649 (2011).

[23] Rocha, M., Sousa, P., Rio, M., Cortez, P.: QoS constrained internet routing with evolutionary algorithms. *In Proceedings of IEEE Congress on Evolutionary Computation*, pp. 2720-2727 (2006).

[24] Cortez, P., Rio, M., Rocha, M., Sousa, P.: Multi-scale Internet traffic forecasting using neural networks and time series methods. *Expert Systems Journal*, Volume 29, Issue 2, pp. 143-155 (2012).

[25] ns-2 (The Network Simulator). Sources and Documentation from http://www.isi.edu/nsnam/ns/.

[26] Eger, K., Hofeld, T., Binzenhofer, A., Kunzmann, G.: Efficient Simulation of Large-Scale P2P Networks: Packet-level vs. Flow-level Simulations. *In Proceedings of 2nd Workshop on the Use of P2P, GRID and Agents for the Development of Content Networks* (2007).

Pedro Sousa graduated in Systems and Informatics Engineering at the University of Minho, Portugal, in 1995. He obtained a MSc Degree (1997) and a PhD Degree (2005), both in Computer Science, at the same University. In 1996, he joined the Computer Communications Group of the Department of Informatics at University of Minho, where he is an Assistant Professor and performs his research activities within Centro Algoritmi at the same university. He is also member of the IEEE professional association. (for additional information and research topics visit http://marco.uminho.pt/~pns)