Network Layer Performance in Peer-to-Peer File Sharing Systems

Raffaele Bolla, Roberto Bruschi, Michele Sciuto
Dept of Communication, Computer and System Sciences,
University of Genoa, Italy
Email: raffaele.bolla, roberto.bruschi, michele.sciuto@unige.it

Krys Pawlikowski
Dept of Computer Science and Software Engineering,
University of Canterbury, New Zealand
Email: krys.pawlikowski@canterbury.ac.nz

***Abstract*—Nowadays P2P applications dominate many networks. However, despite of their diffusion, the analysis of teletraffic generated by these applications in real-time environment remains an open issue. In this paper we look at analytical and simulation models, which can be used for quantifying volumes of P2P teletraffic at the network layer, to ease still existing difficulties associated with monitoring such teletraffic. We present an event-driven simulator, able to simulate P2P teletraffic both at the application and network layers in large overlay networks (we have successfully tested it in networks used by up to 500,000 peers). The simulation model was developed for studying the impact that P2P systems have on the network layer performance. We studied also the application layer dynamics by looking at time evolution of file-downloading processes and at the offered load generated in such networks. We report the results of several simulation scenarios, in which we focused on the consequences such teletraffic has on the Internet access link of an enterprise network.**

***Index Terms*—File-sharing applications; Peer-to-Peer modeling; Network layer; Traffic simulation; Internet traffic.**

I. Introduction

P2P applications still represent today one of the most source of traffic in the Internet. Their large diffusion is testified by many measurement studies, which reveal that such a diffusion is not restrained to home broadband Internet accesses, but also involves enterprise networks [1] - [3] as well as networks of industries and commercial companies [4].

The ability to quantify the impact of P2P traffic on the network is fundamental to a number of network operations, including traffic engineering, capacity planning, quality of service, forecasting for long-term provisioning, etc. The difficulties in estimating P2P traffic volumes lie in the nature of such protocols, which are specifically designed with the aim of hiding their presence in the network [5]. Despite the important efforts of the research community in this direction (see [6] - [8]) P2P traffic detection and classification are still open issues. However, the present issues in measuring and estimating P2P traffic can be faced with traffic models. With this respect, we can cite again [2] and [3], where simulation results are presented, with the intention of studying the gains potentially derived from traffic locality in P2P systems. Many other modeling and simulation works look at P2P overlay networks, combined with the difficulties of simulating the large size overlay networks of real systems [9] - [14]. An interesting approach is the one presented in [9], where a framework for P2P simulation environment, on top of existing packet-level network simulators, has been developed; the underlying layers seem to be considered in detail, even though this approach focuses on the packet level, which could not clearly scale to the size of a real P2P network. In [14] a survey of several P2P simulators revealed the lack of simulators, which could really be used for planning purposes. Well known scalable simulators are PlanetSim, Neurogrid and PeerSim, but they have different aims with respect to our work. In PlanetSim [11], developers can work at two main levels: creating and testing new overlay algorithms like Chord or Pastry, or creating and testing new services (DHT, CAST, DOLR, etc) on top of existing overlays. Neurogrid [12] can simulate many nodes (more than 300,000), but it is specifically developed for the study of application layer [13], not being developed for dealing with network layer traffic. PeerSim can simulate more than 1,000,000 nodes, it results very scalable and accurate for evaluating new P2P protocols at the application layer; however it has not been developed for studying the impact of traffic on the network layer. It is clear that scalability and network layer analysis are not well supported features of existing P2P simulators.

With this problem in mind, we have built an event driven P2P simulation model, able to represent both the application and the network layers and that can be used for simulating large overlay networks (we have successfully tested it with up to 500,000 peers). We focused our attention on the unstructured and decentralized Gnutella overlay network, which still represents an important point of reference in the P2P community. Many recent works on Gnutella network have stimulating our interest ([15], [16]), revealing that the fully decentralized paradigm of Gnutella still represents a reference architecture. As with all decentralized networks, the Gnutella network, such as the most recent Kad network [17], requires no official or common servers. As such, it cannot be disabled by shutting down a given subset of key nodes. Our model has the aim of studying the impact that such P2P systems have on the network layer performance. We start from the application layer, modeling the dynamics of peers, ultrapeers and files in the overlay network and we focus our attention on the results of file queries. We consider a positive query as a file transfer, and then we derive the number of file downloads and the average offered load generated in the network. We do not model the elastic behavior of network traffic and the

978-1-4244-1785-8

dynamic effects of TCP protocol, because we think that a simulation tool like the one presented here can give a useful contribution in network planning, also avoiding a lower level analysis. In fact, download time in file sharing systems can be really influenced by several aspects: network bottlenecks, traffic filters, unexpected user behaviors, reactive mechanisms of the application protocol, protocol incentives for content sharing and others unstable factors. For this reason we do not aspire to describe network performance with high temporal resolutions but we present the P2P traffic offered load as averaged in one hour long time periods. In the following we report the results of several simulation scenarios, in which we focus on the consequences of traffic on the Internet access link of an enterprise network.

The paper is organized as follows. Section II describes the proposed simulation model, including its configuration and validation. Section III analyzes the query success rate of the system, and how it can be influenced by modifications of the overlay network topology. Section IV discusses network performance in different overlay network topologies. In Section V, network performance resulting from the modification of some parameters at the application layer is shown. Finally, the conclusions are reported in Section VI.

II. Model Description

We consider Gnutella 0.6 as a reference system, which relies on a hierarchical unstructured and decentralized overlay network. This choice has different motivations: Gnutella is one of the most used P2P protocols and is well described in literature. Moreover, there are many measurement results in Gnutella networks ([19] - [21]) we found useful both as input data and for validation. A description of the Gnutella protocol can be found in [22].

A. Overlay Network: elements and dynamics

Here we explain how we model the overlay network by considering its elements and its dynamism. The main information which characterizes peers are: connection bandwidth, download queue size and list of files. Connection bandwidth indicates the bandwidth of the access circuit to the network. Download queue size represents the number of download processes, requested to the peer; this parameter, in conjunction with the connection bandwidth, is considered after a query process, in order to choose the best suitable peer for the file download (i.e., the peer with the highest ratio bandwidth/number of downloads). Each peer has a location in the network (see Section II-D); this aspect is important when studying the network/transport layer. In order to participate in the overlay network, each peer needs a connection with an ultrapeer. We consider ultrapeers just for managing the query process and we do not model their deployment in the network. This allows to model just one type of peers, and to consider ultrapeers like external entities responsible for forwarding queries.

We introduced births and deaths of peers in the systems, which cause new connections and disconnections of peers, both at the network and at the application layer. Besides births and deaths of peers, we also model their activation and deactivation. In real P2P systems it is common that a peer is periodically online for limited time intervals. For this reason we assumed it would not be correct to describe these connections and disconnections as they were births and deaths. The events of activation and deactivation preserve memory about the files stored by users and reflect the typical behavior of users in a P2P system: they start the file-sharing application, getting a new connection in the overlay network and, consequently, they try to resume the download and upload processes that were suspended since the previous connections end. In addition, they can also perform new queries.

Our model aims at describing the different behaviors that a peer can assume during day or night hours. We introduced different geographical areas, and the relative time zones, in order to represent the behavior of peers during the whole day. Time zones are relative to four different geographical regions. We approximated such areas as Europe, East US, West US and Asia. Each zone has a different population size, which we set as a configurable parameter. However, as measured and reported in [21], in the following results we uniformly deployed 90% of peers in Europe and North America (East and West), while the remaining 10% in Asia. We consider such different zones to properly model the query generation. For this reason we assumed different query rates during daytime and nighttime. At this stage, we set the same query rate for all users, independently of the geographical zone.

For what concerns the rate of events, the births of peers are driven by an exponential distribution. The expected value λ_{peers} depends on the peers uptime and on the total number of peers in the system, which we assumed to keep constant in time. If we define U as the expected value of the peers' uptime, the births' rate is given by $\lambda_{peers} = N_{peers}/U$. The death of peers is driven by an exponential distribution as well, with expected value equal to the peers uptime. The events of activation and deactivation are also generated by exponential distributions.

B. The file popularity model

We started from the model already presented in [23], in which we investigated the interesting relationship between the Product Life Cycle (PLC) in marketing literature and file popularity in a file-sharing system. In that version of the model, we introduced an analytical function, representing the PLC in marketing environment, for modulating the query interarrival time of files. We represented the query event with reference to each file, and not to the peers, as it is usually done in P2P simulators [14].

In this work the model has been improved, in particular we applied two main modifications with respect to the previous one:

- the popularity curve has been replaced by a family of curves, in order to represent many categories of files, with different popularities;
- the query process is now driven by peers and not by files.

We assume the presence of two different types of files, as suggested by [24]. The files already present in the system are considered as having a constant popularity. They are modeled as "old files", i.e. belonging to the system for a long time. Every time a peer decides to share a new content, a new file joins the network during the simulation. These files are considered as *new* and they are characterized by a popularity which dynamically changes with the time. This means that the interest they generate in the system is heavily time dependent. As reported in [19] file popularity has a life cycle of one year, and it can change with a daily granularity. We assumed that a file could join the network at every instant of its PLC. This means it can appear in the system in the first or in the final stage of its popularity cycle. This temporal instant, describing the position of the file in its PLC at the time of its birth, is defined as T_{pop_0}. Every time a birth of file occurs, T_{pop_0} is randomly chosen in a set of 365 integer numbers. These numbers represent the number of days in a time period of one year.

We considered the following family of curves, as suggested in [18]

$$\Psi_K(\tau) = \begin{cases} (e^{\tau/K} - 1)/(e-1) & \text{if } \tau \in [0, K) \\ (K-1)/\tau & \text{if } \tau \in [K, \infty) \end{cases} \quad (1)$$

where $K = 30 \cdot c$ and $c = 1, \ldots, 6$. K indicates the maximum of popularity for the c^{th} function. The family of curves, for $\tau \in [0, 365]$, is shown in Fig. 1.

We considered 6 different curves of the same family, with adjacent popularity peaks shifted of 30 days between them. The *new* files are assigned with uniform probability to one of the 6 categories. Each category is characterized by a popularity function, which influences the number of queries received by its files. Every time a peer emits a query, the requested k^{th} file is chosen with a probability depending on

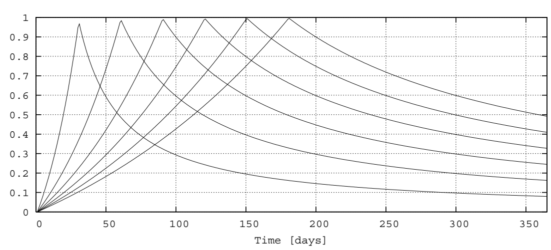

Fig. 1. Family of popularity curves

$$\frac{\pi_k}{\sum_{i=0}^{N-1} \pi_i}, \quad (2)$$

where π_k is its popularity (the value assumed by the popularity function) and N is the total number of files. So, during the simulation the queries for the new files follow the popularity function.

For the sake of scalability we assumed that peers do not emit single queries but bursts of them. At the time of a peer connection to the overlay network, a burst of queries is generated; then, during its active period, successive bursts are performed (driven by an exponential distribution). The burst rate has two states: during the nighttime we scaled it down by a factor of 10, with respect to the daytime. When a burst of queries is generated, the number of queries q is chosen in $[0, q_{max}]$ with uniform probability, where q_{max} is a configurable parameter.

C. The popularity model effects on file queries

We analyzed the effects of such a popularity model in the number of queries received by files with different popularities. In Fig. 2 we plotted the popularity behavior for three different files along with the number of queries received by each of them over a whole year. The behavior of the first two files derives from a dynamic simulation. We also reported an interesting effect, performed in case of a static simulation (without births and deaths of peers). Without the births and deaths of peers, the plotted file reaches a saturation in the number of queries, because at the 200-th simulation day it is already kept by the most part of peers in the network (and then nodoby is interested in it anymore).

For the sake of simplicity, the results of this section have been performed in case of small networks, with 2,700 peers.

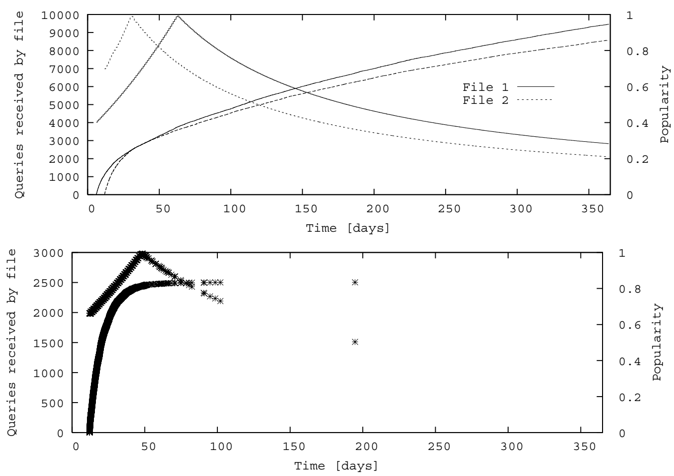

Fig. 2. Number of queries and popularity behaviors for two different files in a dynamic simulation in 2(a) and for one file in a static simulation (without peers' dynamism) in 2(b)

D. Network Layer

We consider the n access nodes as routers of an enterprise network, where one of them provides the access link to the Internet. We simulate the whole Internet behind the $(n-1)^{th}$ node. The traffic matrix is computed by considering traffic between the n access nodes. Every time a query process can reach the end, a file download is counted between the peers of the overlay network, and consequently we assume that a traffic flow will start between two access nodes of the network. We define the traffic matrix $D = (d_{i,j})_{n \times n}$ for all $0 \le i, j \le n-1$, where

- n is the number of access nodes,
- i is the access node of the downloading peer and
- j is the access node of the uploading peer.

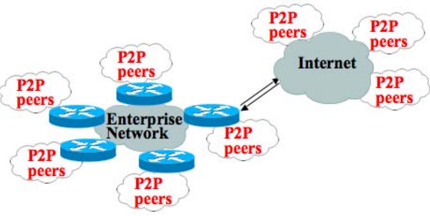

Fig. 3. Network scenario

So, the element $d_{i,j}$ represents the number of downloads occurred between the downloading peer connected to the i^{th} node and the uploading peer connected to the j^{th} node. This means that the traffic flows from the j^{th} node to the i^{th} node. From the traffic matrix, we can derive the time evolution of file transfers considering the following elements:

- $d_{n-1,n-1}$ is the traffic in the big Internet;
- $\sum_{i,j=0}^{n-2} d_{i,j}$ is the total traffic inside the network;
- $\sum_{i=0}^{n-2} d_{i,n-1}$ is the total traffic going into the network through the Internet connection;
- $\sum_{j=0}^{n-2} d_{n-1,j}$ is the total traffic going out the network through the Internet connection.

In the following we focus our interest on the traffic flowing through the access link to the Internet, in both directions.

E. Parameters and configuration

We have designed the model and the simulator with the aim of studying the file-sharing traffic and its impact on the network layer. In order to analyze many different case studies, we have developed a software tool which allows the configuration of several parameters. In TABLE I we report all configurable parameters we can change in order to test different scenarios. N_{peers}, N_{files} and N_{ultrap} are the number of peers, files and ultrapeers at $t = 0$. In order to control the concentrations of peers in some places of the overlay network, we introduced a parameter α expressing the number of leaves for every ultrapeer. The overlay mesh index μ defines the maximum number of neighbors for every ultrapeer, expressed by $N_{neighbors} = \mu \cdot N_{ultrapeers}$. Different values of α and μ modify the overlay network topology. The first one changes the size of clusters of peers connected to ultrapeers, while μ controls the number of links in the first layer of the overlay network. The dimensions of files are defined by δ_i ($i = 1, 2, 3$), for three different size classes. The percentage of each category δ_i is defined by P_{δ_i}. The number of shared files by peers is represented by ϕ_i ($i = 1, 2, 3$). P_{ϕ_i} means the percentage of peers sharing ϕ_i files. We have also considered the possibility to define three different connection bandwidths for peers, the relative parameter are β_i ($i = 1, 2, 3$) and P_{β_i}, representing the percentage of peer with the same access bandwidth to the network. Q is the number of query hops in case of fault, its default value is 2. Changing the Q value we can limit the depth of queries to their own cluster ($Q = 0$) or we can allow ultrapeers to flood queries to their neighbors ($Q = 1$) or to neighbors of their neighbors ($Q = 2$). In order to control the attitude of peers in sharing files, we introduced a parameter F, representing the probability of sharing downloaded files. Every time a file is downloaded, the probability that such a file will be shared is determined by F. The number of shared files is upper bounded by ϕ_i. U is the expected value for peer' uptime, U_{ac} determines the duration of the activity periods of peers in their uptime, BF is the rate of births of files in the system and λ_q is the emission rate of the bursts of query.

N_{peers}	Initial number of peers
N_{files}	Initial number of files
N_{ultrap}	Initial number of ultrapeers
α	Number of leaves for every ultrapeer
μ	Mesh index
δ_i, P_{δ_i}	Dimension of files and % of files, in each category $i = 1, 2, 3$.
ϕ_i, P_{ϕ_i}	Number of files shared by peers and % of peers, in each category $i = 1, 2, 3$.
β_i, P_{β_i}	Connection bandwidth of peers and % of peers in each category $i = 1, 2, 3$.
Q	Number of query hops
F	Percentage of files shared by peers
U	Average peers' uptime
U_{ac}	Average active (inactive) period for peers
BF	Birth rate of files
λ_q	Rate of query bursts

TABLE I
CONFIGURATION PARAMETERS AT THE APPLICATION LAYER

F. Initialization Procedure

The initialization procedure provides the full topology of the overlay network and the assignment of files to peers. The overlay network is initialized by considering the configuration parameters in TABLE I, and with the constrain that every ultrapeer needs at least one neighbor, in order to provide connection for every peer in the network. It is assumed that each peer can maintain no more than one copy of each file and that it must have one relationship with an ultrapeer. The initial number of replicas is a specific parameter associated with each different file. We assign this parameter, according to the Zipf distribution, as reported in [25].

III. EFFECTS ON THE QUERY SUCCESS RATE OF CHANGES IN THE OVERLAY NETWORK TOPOLOGY

Before the section concerning the network layer, we report results about the query success rate of the system, showing that it can be strongly biased by modifications in the topology of the overlay network. In Fig. 4 we report the ECDF of the query success rate in case of 4 different topologies.

We first set the network parameters as follows: 27,000 peers, 1,000 ultrapeers and 16 neighbors per ultrapeer on average (α = 27, μ = 0.016). These parameters reflect the real overlay network topology, reported in many measurement works, such as [15], [16], [26], [27]. The query success rate, deriving from this type of configuration, results comparable with the one published by Qiao and Bustamante in [15]; we can consequently claim that the model represents accurately the query mechanisms of the overlay network.

We also report the case of an increment in the number of ultrapeers (up to 2,700, with α = 10 and μ = 0.006) which raises the profile, reducing the query success rate. The reason lies in the higher number of clusters (system composed by an ultrapeer with its leaves) with less leaves attached, which consequently maintains a lower number of information about the location of contents.

In case of a lower number of ultrapeers (270), with still 16 neighbors each (α = 100, μ = 0.06), we verified that the query success rate results strongly improved. Here we are in presence of just 270 clusters, with a high number of leaves. Even if the query success rate is extremely high, the overlay network results less resilient; e.g. in case of the death of one ultrapeer, around 1,000 leaves have to find one new ultrapeer, with a consequent instability in the overlay. Finally we show the case of 270 ultrapeers with just 5 neighbors (α = 100, μ = 0.0185). Here the low number of neighbors slows significantly down the query success rate, highlighting the importance in the number of neighbors at the first hierarchy layer. The higher is the number of neighbors, the higher is the number of contacted ultrapeers during the query process, with a consequent increment in the query success rate.

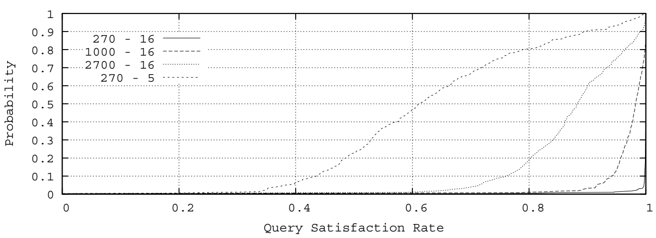

Fig. 4. ECDF of the query success rate for different overlay network topologies. The number of ultrapeers and their number of neighbors are reported for each curve

IV. DIFFERENT OVERLAY NETWORK TOPOLOGIES AND THEIR IMPACT ON THE NETWORK LAYER

We conducted simulations for understanding the type of traffic resulted from P2P file sharing applications and its influence on the network layer. We present results, which illustrate how our simulator can be used.

Note that, since our proposal aims at studying the transient dynamics of traffic profiles in P2P systems, we do not report results of simulations at the steady-state, but we rather focus our attention on the evolution of traffic over fixed time intervals, in different scenarios.

For this purpose we consider the evolution of the traffic matrix, counting the number of occurred downloads every fixed time interval, and consequently the average offered load generated by network nodes. We take into account the distribution of file sizes measured in Gnutella network by Reza et al. [19], in order to choose the right values of δ_i and P_{δ_i}. These results say that the 6% of files is composed of videos while the 94% of files represents audio or other types of files (.txt, .exe, images...). Although there is a big gap in the number of files between these two categories, their total amount in bytes is equally distributed. The video files have an average value of 500 MBytes, while the mean value for the other category is 30 MBytes. With respects to the cited measurements, in the following examples we consider an average value of 60 MBytes for the sizes of files, with the aim of providing and estimation of the average traffic load, deriving from the number of downloads.

The number of shared files by peers (ϕ_i and P_{ϕ_i}), in accordance with measurement results, has been set as follows: 90% of peers shares less than 100 files; only 1% of the total makes available more than 1,000 files (up to 5,000); the remaining part, counting less than 10%, shares a number of contents ranging from 1,000 to 5,000.

For the sake of simplicity we present the traffic matrix evolution considering an overlay network with 27,000 peers, 1,000 ultrapeers and 500 files (at the beginning of the simulation). Where not specified, Q = 2, F = 50%, α = 27, μ = 0.016, BF = 1.157 $\cdot 10^{-5}$, q_{max} = 10, U = 60 days, U_{ac} = 48 hours, λ_q = 0.463 $\cdot 10^{-4}$ (please, note that this is an emission rate of bursts and not of single queries). We want to study a particular case of an enterprise network (EN) connected to a public network (which could represent a portion of the whole Internet). We simulate such a scenario displacing 25,000 peers behind one network node, which represents the connection of the EN to the public network, and connecting the remaining peers to the other access nodes (intranet) with a uniform distribution. We want to study the time evolution of the following types of traffic: the total traffic among the access links of the EN and the total traffic flowing through the connection to the public network, in both directions. We just limit the analysis to P2P traffic, neglecting other applications. We also assume that all the network links are overprovisioned and that system performance are not influenced by potential bottlenecks. The results always concern the same network layer, composed by 8 access nodes.We simulate the whole Internet behind the node 7. From the traffic matrix, we can derive the time evolution of traffic considering the following elements:

- $\sum_{i,j=0}^{6} d_{i,j}$ is total the traffic inside the network;
- $\sum_{i=0}^{6} d_{i,7}$ is the total traffic going into the network through the Internet connection;
- $\sum_{j=0}^{6} d_{7,j}$ is the total traffic going out the network through the Internet connection.

A. An enterprise network with an Internet connection

In Fig. 5 we report the behavior of the different categories of traffic. The traffic rate in the access link has different profiles in both directions. The download profile (IN) follows a periodic behavior, due to the major rate of queries performed by peers during the day. The upload curve (OUT) has a more smooth profile which evidences the strong effects on the network of P2P file sharing traffic, even in absence of direct users' interactions. Inside the EN the average total traffic rate has a periodic profile, which reveals a file sharing activity also among the internal peers.

Fig. 6 reports the results of the analog situation, where the enterprise network has a double size with respect to the previous case in Fig. 5. The trends of the traffic profiles find a confirmation, but it is interesting to highlight that the traffic rate in the access link to the public network results less than the double of the previous one (in Fig. 6), while the internal traffic is strongly biased by the increased number of peers. This figure provides the interesting evidence that in case of an increasing number of peers the first consequence is a strong file sharing activity between them, which reduces the impact of traffic on the access link to the public network. We reported the average number of downloads in TABLE II, where it is clear the different impact of traffic in the two cases.

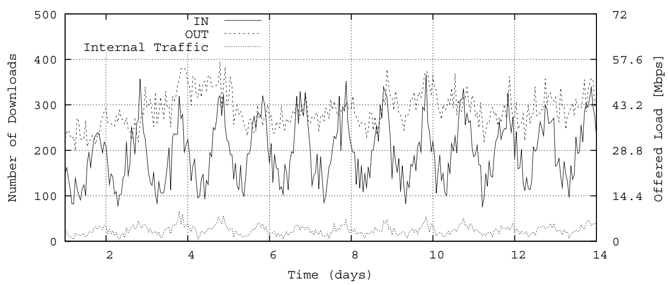

Fig. 5. Offered load in case of 2,000 peers in the EN

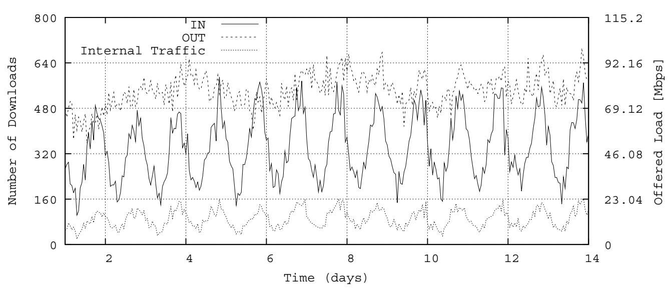

Fig. 6. Offered load in case of 4,000 peers in the EN. The number of peers in the public network is the same as in previous case

	IN	OUT	Internal Traffic
2,000 peers	192	291	24
4,000 peers	334	550	88

TABLE II
AVERAGE NUMBER OF DOWNLOADS

B. An enterprise network with an Internet connection and a honey-pot with 100 popular files shared suddenly by one peer

In this section we consider the presence of a peer which appears in the enterprise network sharing 100 files. All the files have low popularity values, characterized by T_{pop_0} = 364. The network has the same features of the one described in Fig. 5. The 100 popular files are shared by the peer instantaneously after 48 hours of simulation time. This is a realistic case, where

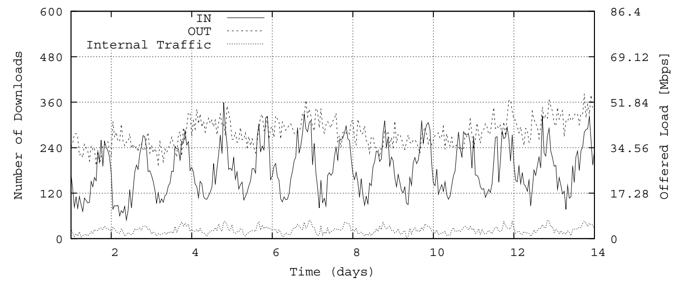

Fig. 7. Offered load in case of a honey-pot birth in the EN. Here the overlay network has 1,000 ultrapeer

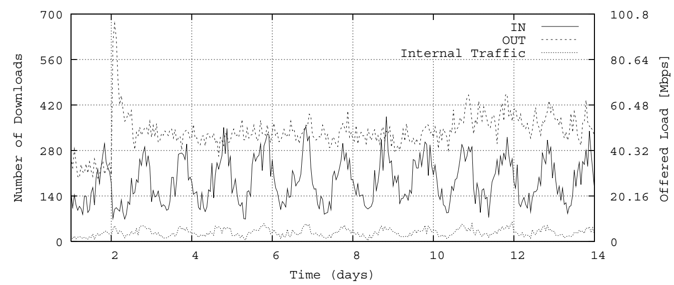

Fig. 8. Offered load in case of a honey-pot birth, generating a flash crowd in the network. Here the overlay network has 270 ultrapeer

P2P traffic rate increases for the intense activity of one peer sharing a large set of files.

We run simulations in several overlay networks, and we realized that the sudden presence of a honey-pot can have different impacts on the network traffic. In Fig. 7, the birth of the honey-pot does not have important effects on the traffic; it is possible to notice just a short gap in the upload profile at the beginning of the second day. Here we are in the case of 1,000 ultrapeers.

Considering Fig. 8 (where 270 ultrapeers are present) is possible to verify the effects of such an event on the traffic. At the instant of the peer's birth the upload traffic profile reports a spike, which evidences the requests for the 100 files coming from the public network. The traffic strongly increases, reaching a peak as high as around three times the previous value of traffic. After the birth of files, the outgoing traffic does not rise anymore, reporting a constant behavior. Initially files are available just in the EN, but as soon as they are downloaded, they become available in the whole network, causing a decline in the peak profile. The differences in Figures 7 and 8 are caused by the different query success rates of two overlay network topologies, reported in Fig. 4. In case of a low number of ultrapeers, the query success rate is so high that the presence of a honey-pot has an immediate impact on the network. In a more realistic case, like the one in Fig. 7, the strong sharing activity of one peer does not change significantly the traffic profiles.

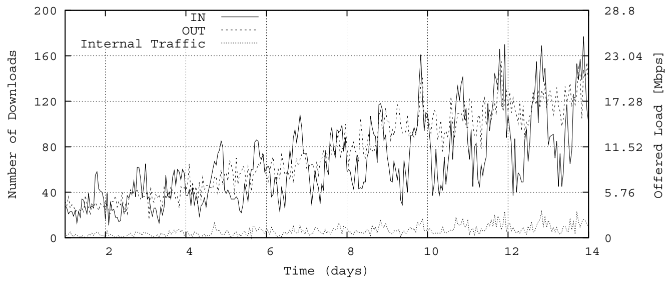

Fig. 9. Offered load in case of Q = 1

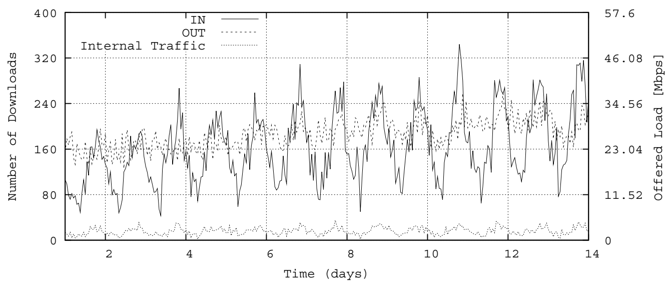

Fig. 10. Offered load in case of F = 1

V. Effects on the network caused by application layer modifications

In this Section we look at our findings related with network performance at the application layer. The network scenario is the same presented in Fig. 5. We modified the value of queries' depth Q and the probability F that peers share a downloaded file. Fig. 9 shows the behavior of traffic for Q = 1, i.e. if queries which do not get results in their own cluster are just forwarded to the first ring of ultrapeers. The second hop is not performed. We expect to find a lower amount of traffic rate, due to the higher query fault probability. Looking at Fig. 9 (with respect to Fig. 5, where the same network in case of Q = 2 is presented), we find an increasing behavior in the traffic profiles which can be explained considering the lower query success rate and consequently the slower diffusion of files in the network.

Figures 10 and 11 show the results for F equal to 1% and 100%. Such a scenario could interest a network provider, who wishes to understand the amount of traffic derived from different users' behaviors in sharing contents. Obviously we would expect to verify a traffic decrease in case of F = 1% and an increment in case of F = 100%, which means that peers share all the downloaded contents. Note that the simulations report lower values of traffic (with respect to Fig. 5, where the same network in case of F = 50 % has been studied) in the first case with F = 1%, while almost negligible effects in the case with F = 100%. The reason lies in the low query success rate in case F = 1%. For F = 100% the query success rate does not result influenced, giving evidence that a peers' sharing activity of the 50% is already enough for providing a consistent file sharing traffic.

We also considered modifications of the overlay network topology, with a focus on the ratio between the number of

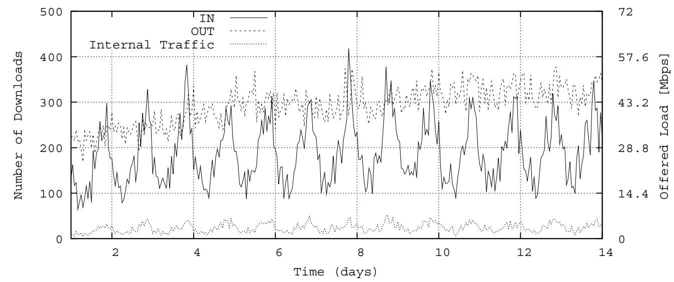

Fig. 11. Offered load in case of F = 100

peers and ultrapeers in the system (expressed by α) and in the number of neighbors for each ultrapeer (expressed by μ). In the previously shown results, we considered an overlay network composed by 27,000 peers, 1,000 ultrapeers (α = 27) and 16 neighbors per ultrapeer on average. As explained in Section III, such a topology reflects correctly the query success rate measured in the real Gnutella network, and it has determined our choices in defining the configuration parameters of the simulator. In the following we show the effects on the traffic of α and μ modifications.

We first show the case of 2,700 ultrapeers and 16 neighbors in Fig. 12. We would expect a reduction in the traffic with respect to the reference traffic profiles in Fig. 5, due to the lower query success rate reported in Fig. 4. The figure evidences that the system needs some days to reach the same values of offered load. The effect of the reduction in the query success rate is a slower trend in reaching the same traffic volume. We can then claim that the traffic generated in a network with a high number of ultrapeers (2,700) does not report differences, the offered load is the same but it is reached in longer temporal intervals.

Figures 13 and 14 reveal that a low number of ultrapeers' neighbors strongly decreases the traffic load. In both the cases a steady state is reached in long time periods, because the query success rate is importantly decreased by the low number of ultrapeers and neighbors.

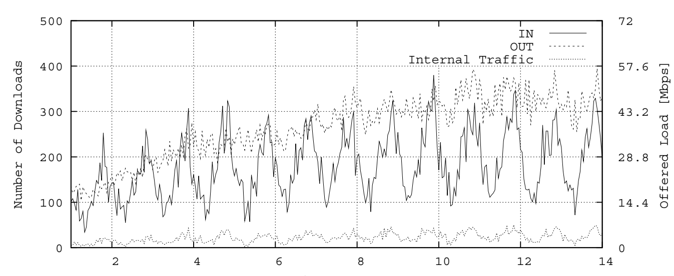

Fig. 12. Offered load in case of 2,700 ultrapeers with 16 neighbors (μ = 0.006)

VI. Conclusions

The presented model was developed for studying the possible impact of large P2P systems on the network layer's performance. We studied also the application layer dynamics by looking at time evolution of numbers of downloaded files

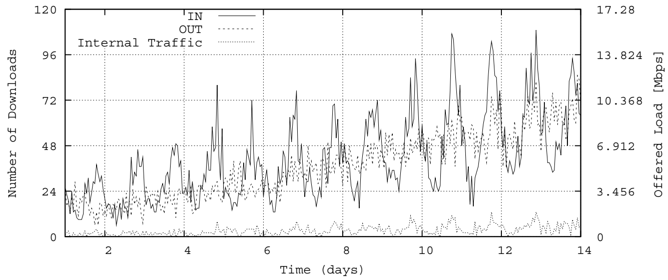

Fig. 13. Offered load in case of 1,000 ultrapeers with 5 neighbors (μ = 0.005).

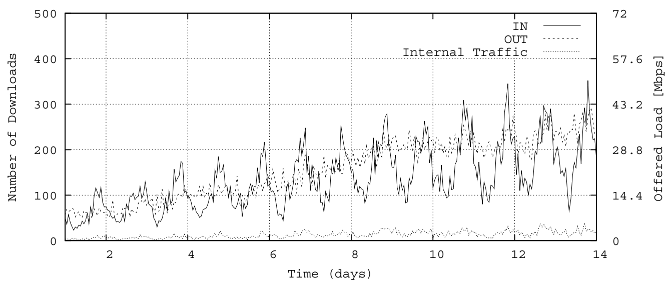

Fig. 14. Offered load in case of 270 ultrapeers with 5 neighbors (μ = 0.0185)

and the offered load generated in such networks. We have shown interesting effects in presence of different scenarios, with a set of results which evidences how P2P traffic can impact on the network. Future extensions of the work will face the issue of how to design a network infrastructure taking into account the effects of P2P traffic. We also intend to study more precise traffic generation models, to allow more accurate analysis of throughput generated by P2P applications.

REFERENCES

[1] M. Yang, Y. Dai, and J. Tian. Analyzing Peer-to-Peer Traffic's Impact on Large Scale Networks. In Proc. of ICCS 2006, pp. 412-419.

[2] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, J. Zahorjan. Measurement, Modeling and Analysis of a Peer-to-Peer File-Sharing Workload. In Proc. of the 19th ACM Symposium of Operating Systems Principles (SOSP), Bolton Landing, NY, October 2003.

[3] Thomas Karagiannis, Pablo Rodriguez, Dina Papagiannaki. Should Internet Service Providers Fear Peer-Assisted Content Distribution? Internet Measurement Conference (IMC), Berkeley, CA, USA, October, 2005.

[4] N. Ben Azzouna and F. Guillemin. Experimental analysis of the impact of peer-to-peer applications on traffic in commercial IP networks. European transactions on Telecommunications: Special issue on P2P networking and P2P services, ETT 15(6), November-December 2004.

[5] T. Karagiannis, A. Broido, N. Brownlee, K. C. Claffy, M. Faloutsos. Is P2P dying or just hiding? In Proc. of the GLOBECOM 2004 Conference, IEEE Computer Society Press, Dallas, Texas, November 2004.

[6] T. Karagiannis, K. Papagiannaki and M. Faloutsos. BLINC: Multilevel Traffic Classification in the Dark. In Proc. of the 2005 Conference on Applications, technologies, architectures, and protocols for computer communications. ACM SIGCOMM 2005, Philadelphia, USA, pp.229-240.

[7] M. Crotti, F. Gringoli, P. Pelosato, L. Salgarelli. A statistical approach to IP-level classification of network traffic. In Proc. of the 2006 IEEE Int. Conference on Communications, June, 2006.

[8] A. Moore and K. Papagiannaki. Toward the accurate identification of network applications. In Passive & Active Measurement Workshop, Boston, USA, March 2005.

[9] Q. He, M. Ammar, G. Riley, H. Raj, R. Fujimoto. Mapping Peer Behavior to Packet-level Details: A Framework for Packet-level Simulation of Peer-to-Peer Systems. In Proc. of IEEE/ACM Int. Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems (MASCOTS03), Atlanta, USA, October 2003, pp.71-78.

[10] PeerSim. http://peersim.sourceforge.net/

[11] P. García, C. Pairot, R. Mondéjar, J. Pujol, H. Tejedor, R. Rallo. PlanetSim: A New Overlay Network Simulation Framework. Lecture Notes in Computer Science (LNCS), Vol. 3437. March 2005, pp. 123-137.

[12] NeuroGrid. http://www.neurogrid.net/php/index.php

[13] Sam Joseph. An Extendible Open Source P2P Simulator. P2PJournal, 2003

[14] S. Naicken, A. Basu, B. Livingston, S.Rodhetbhai, I. Wakeman. Towards Yet Another Peer-to-Peer Simulator. In Proc. of the Forth Int. Conference in Performance Modelling and Evaluation of Heterogeneous Networks (HET-NETs '06), West Yorkshire, U.K., September 2006.

[15] Y. Qiao and F. E. Bustamante. Structured and unstructured overlays under the microscope: a measurement-based view of two P2P systems that people use. In Proceedings of the Annual Technical Conference on Usenix'06, Boston, MA, May 30 - June 03, 2006

[16] Ho-Hyun Park, Woosik Kim, Miae Woo A Gnutella-based P2P System Using Cross-Layer Design for MANET International Journal of Electronics, Circuits and Systems (IJECS), Vol. 1 Num. 3, 2007, pp. 139 -144

[17] P. Maymounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In Proceedings of IPTPS02, Cambridge, USA, Mar. 2002.

[18] Michele Garetto, Daniel Figueiredo, Rossano Gaeta, Matteo Sereno. A Modeling Framework to Understand the Tussle between ISPs and Peer-to-Peer File Sharing Users. In Performance Evaluation, Vol. 64, Num. 9-12 (Oct. 2007), pp. 819-837.

[19] D. Stutzbach, S. Zhao, R. Rejaie. Characterizing Files in the Gnutella Network. Multimedia Systems Journal, 2007.

[20] D. Stutzbach, R. Rejaie. Understanding Churn in Peer-to-Peer Networks. In Proc. of ACM SIGCOMM/USENIX Internet Measurement Conference, Brazil, October 2006.

[21] A. Klemm, C. Lindemann, M. K. Vernon, O. P. Waldhorst. Characterizing the query behavior in peer-to-peer file sharing systems. In Proc. of ACM SIGCOMM Conference on Internet measurement, Italy, October 2004, pp. 55-67.

[22] RFC-Gnutella. http://rfc-gnutella.sourceforge.net

[23] R. Bolla, M. Eickhoff, K. Pawlikowski, M. Sciuto. Modeling file popularity in peer-to-peer file sharing systems. In Proc. of the 14th Int. Conference on Analytical and Stochastic Modelling Techniques and Applications, pp. 149 - 155. Prague, Czech Republic, June 2007.

[24] T. Hoßfeld, K. Leibnitz, R. Pries, K. Tutschku, P. Tran-Gia, K. Pawlikowski. Information Diffusion in eDonkey Filesharing Networks. In Proc. of Australian Telecommunication Networks and Application Conference (ATNAC04), Sydney, Australia, December 2004, pp. 390-397.

[25] T. Schlosser, T. E. Condie, S. D. Kamvar. Simulating A File-Sharing P2P Network. In Proc. of the 1st Workshop on Semantics in P2P and Grid Computing, Budapest, Hungary, May 2003.

[26] Boon Thau Loo, Joseph M. Hellerstein, Ryan Huebsch, Scott Shenker, and Ion Stoica. Enhancing P2P File-Sharing with an Internet-Scale Query Processor. In Proc. of the 30th International Conference on Very Large Data Bases, September 2004.

[27] Boon Thau Loo, Ryan Huebsch, Ion Stoica, and Joseph M. Hellerstein. The Case for a Hybrid P2P Search Infrastructure. In Proc. of the 3rd International Workshop on Peer-to-Peer Systems (IPTPS), San Diego, CA, February 2004. Berkeley, June 2003.