Identification and Analysis of Peer-to-Peer Traffic

Marcell Perényi, Trang Dinh Dang, András Gefferth, Sándor Molnár
Budapest University of Technology & Economics, Department of Telecommunications & Media Informatics, Budapest, Hungary
Email: {perenyim, trang, gefferth, molnar}@tmit.bme.hu

***Abstract* – Recent measurement studies report that a significant portion of Internet traffic is unknown. It is very likely that the majority of the unidentified traffic originates from peer-to-peer (P2P) applications. However, traditional techniques to identify P2P traffic seem to fail since these applications usually disguise their existence by using arbitrary ports. In addition to the identification of actual P2P traffic, the characteristics of that type of traffic are also scarcely known.**

The main purpose of this paper is twofold. First, we propose a novel identification method to reveal P2P traffic from traffic aggregation. Our method does not rely on packet payload so we avoid the difficulties arising from legal, privacy-related, financial and technical obstacles. Instead, our method is based on a set of heuristics derived from the robust properties of P2P traffic. We demonstrate our method with current traffic data obtained from one of the largest Internet providers in Hungary. We also show the high accuracy of the proposed algorithm by means of a validation study.

Second, several results of a comprehensive traffic analysis study are reported in the paper. We show the daily behavior of P2P users compared to the non-P2P users. We present our important finding about the almost constant ratio of the P2P and total number of users. Flow sizes and holding times are also analyzed and results of a heavy-tail analysis are described. Finally, we discuss the popularity distribution properties of P2P applications. Our results show that the unique properties of P2P application traffic seem to fade away during aggregation and characteristics of the traffic will be similar to that of other non-P2P traffic aggregation.

***Index Terms* – Peer-to-peer, identification, traffic analysis, heuristics**

I. Introduction

From the beginning of the new millennium the Internet traffic characteristics show a dramatic change due to the emerging *Peer-to-Peer* (P2P) applications. Starting from the first popular one (Napster) a number of new P2P based multimedia file sharing systems have been developed (FastTrack, eDonkey, Gnutella, Direct Connect, etc.). The traffic generated by these P2P applications consumes the biggest portion of bandwidth in campus networks, overtaking the traffic share of the World Wide Web [6, 31]. A common feature in all of these P2P applications is that they are built on the P2P system design where instead of using the server and client concept of the web each peer can function both as a server and a client to the other nodes of the network. This principle involves the adapting nature of P2P systems as individual peers join or leave the network. Another common feature of these P2P systems is that they are mainly used for multimedia file sharing (movies, music files, etc.), which frequently contain very large files (megabytes, gigabytes) in contrast to the typical small size of web pages (kilobytes). A number of studies have been published in the field of P2P networking. Papers [1, 2, 3, 4, 5, 6, 7, 8, 29] focus on the measurement of different P2P systems like Napster, Gnutella, KaZaA, and the traffic characterization and analysis of P2P traffic providing some interesting results of resource characteristics, user behavior, and network performance. Several analytic efforts to model the operation and performance of P2P systems have been presented so far. Queuing models are applied in [9, 10], while in [11, 12, 13] branching processes and Markov models are used to describe P2P systems in the early transient and steady state. P2P analysis using game theory is presented in [23, 24], among others. Other studies, e.g. [14, 15, 16, 17], are concerned with the effective performance and the QoS issues of P2P systems. In addition, many papers [18, 19, 20, 21, 22] indicate various possible applications using P2P principles. Further approaches propose structured P2P systems using Distributed Hash Table (DHT) with several implementations like Pastry, Tapestry, CAN, Chord [25]. The P2P traffic characteristics are not fully explored today and there is a tendency that they will be even more difficult to analyze. In contrast to the first generation P2P systems the recent popular P2P applications disguise their generated traffic resulting in the problematic issue of *traffic identification*. The accurate P2P traffic identification is indispensable in traffic blocking, controlling, measurement and analysis. However, the issue is touched upon in only a few papers and the proposed solutions still have some drawbacks. The problem is that P2P communications are continuously changing, from TCP layers using well-

Based on the “On the Identification and Analysis of P2P Traffic Aggregation”, by Trang Dinh Dang, Marcell Perényi, András Gefferth, and Sándor Molnár which appeared in the Proceedings of 5th International IFIP-TC6 Networking Conference, Coimbra, Portugal, May, 2006. © 2006 IFIP.

known ports in some first versions to both TCP/UDP with arbitrary and/or jumping ports nowadays. A robust and accurate P2P traffic identification is vital for network operators and researchers but today there is a lack of published results on this field and this is our main motivation for the work presented in this paper. The workload characteristics of peers participating in some P2P systems has been examined in several papers as mentioned before. However, from the aspect of service providers only little useful information can be gained from these studies. The service providers are less interested in the detailed activities of some particular P2P softwares but the traffic generated by peer users. This paper concentrates on those factors and characteristics of P2P communications which have an impact on the P2P traffic aggregation. The rest of the paper is organized as follows: we describe our measurements and the pre-processed data in Section II. Section III presents our heuristic P2P identification method, which is verified in Section IV. The traffic identification results are given in Section V, while characterization results in Section VI. Finally, Section VII concludes the paper.

II. TRAFFIC MEASUREMENT

The measurements were taken at one of the largest Internet providers in Hungary in May 2005. The scenario of the measurements is depicted in Fig. 1. In the chosen network segment, traffic of ADSL subscribers is multiplexed in some DSLAMs before entering the ATM access network. Placed at the border of the access network and the core network are some Cisco routers. NetFlow measurements are carried out at two of these routers in three days from May 26th to 28th. NetFlow, developed by Cisco, collects all *inbound* and *outbound* flow information and exports the logs periodically. Some packet-level information was also recorded, including packet arrival times and packet sizes. The obtained data traces are the aggregate incoming traffic of more than 1000 ADSL subscribers.

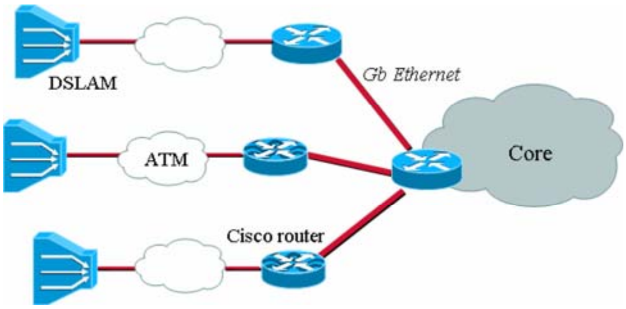

Fig. 1. Location of NetFlow measurements

Two data sets were selected for analysis, which are denoted by *Callrecords 1* and *Callrecords 2*. In the original conference paper the analysis of three other data sets can be found. (The previous three data sets contained only one-way flow-level information and no packet level information.) The summary of the data sets is presented in Table 1.

Data sets	Time of measurement	Number of flows	Total traffic (GB)
Callrecords 1 inbound	22nd July 2005	11 423 510	457.84
Callrecords 1 outbound	22nd July 2005	12 373 446	93.95
Callrecords 2 outbound	4th April 2006	19 057 097	175.62

Table 1. Summary of collected data sets
(the inbound direction of *Callrecords 2* was not processed)

III. P2P TRAFFIC IDENTIFICATION

In general, the issue of application identification inside the IP network is not trivial. This is even more complicated and difficult in the case of P2P applications. Early P2P systems often use TCP with some fixed ports for communication. In these cases the traditional port-based traffic monitoring and classification can be used to measure P2P application traffic. Nowadays the dramatic growth of P2P usage accompanied by the huge bandwidth consumption, together with the problematic content copyright concerns lead to some interventions from network operators such as traffic limiting or blocking. To overcome these limitations newer P2P applications can use both TCP and UDP connections with arbitrary ports for messaging and data transmission. These improvements make the detection of P2P traffic a challenge. This section discusses in detail the P2P identification issue involving previous work and our proposed method.

Discussion of the issue and previous works

Concerning related work we overview a few papers dealing with the issue. A method based on port properties is presented in [31]. The authors note that a substantial number of flows cannot be identified by the mapping method from flows to applications. They classify the unknown flows by size and assume that the traffic is P2P if the flow transmits more than 100kB in less than 30 minutes.

In [26] P2P traffic is identified based on the application signatures found in the payload of data packets. Authors showed that typical sets of strings are identified in the packet payload generated by some P2P applications. The method can be implemented for online tracking of P2P traffic by examining several packets in each flow. It is reported that the technique works with very high accuracy. It seems that the signature-based method can provide the most accurate P2P traffic detection. This method could be used in traffic investigation of one or several particular P2P systems. However, there are also some drawbacks. The very first challenge is the lack of openly available, up-to-date, standard, and complete P2P protocol specification [26]. Since P2P protocols are continuously developed the traces of today will not surely exist in tomorrow's traffic. Furthermore, an increasing number of P2P protocols rely on encryption, so payload matching cannot be applied in these cases.

A similar payload-based method is presented in [28]. This paper also proposes two heuristics for identification of P2P traffic without payload examination. It is reported that more than 90% of the results provided by the payload method is identified by the proposed heuristics. It should be mentioned that the payload examination only tries to detect the traffic generated by several P2P applications. We cannot know for sure all possible P2P applications people use. Nevertheless, the idea is very promising. The identification of P2P traffic aggregation should be done by heuristics which are based on some common properties of P2P communications instead of examining particular P2P applications.

The method described in [32] also works without payload information. Besides flow identification by ports it proposes the estimation of unknown traffic by relating it to preceding, known traffic. The authors argue that traffic induces other traffic so there is a possibility to identify unknown traffic which was induced by known traffic. Since this principle cannot guarantee the correct identification some additional statistics are also used to increase accuracy in the decision method.

Kim *et al.* in [26] provide a method which is an improvement of the network port-based application detection. Their main idea is to discover the relationships between flows that belong to a particular P2P application and then use this information to put measured flows into groups. Flow groups together with a set of typical P2P application ports are used to determine whether a group of flows is generated by P2P applications or not. The disadvantage of this method is that it is very difficult to find appropriate typical relationships between flows of a given P2P application. In addition, as presented in the paper, there is still more than 40% of the total traffic which cannot be identified.

An identification system for pure P2P applications is given in [33]. The method is specialized for the Winny application, which is currently the most popular P2P application in Japan. It uses the server/client relationships among peers. Some evaluation results of the method are also presented.

For the sake of completeness the crawl-and-probe method [3] should also be mentioned. Authors periodically “crawled” the P2P system to gather instantaneous snapshots of a subset of user population and then sent probes to users to directly measure some users’ properties. This method cannot collect users’ traffic activities.

In summary, P2P traffic identification has two promising approaches:

- P2P traffic identification based on payload information
- P2P traffic identification based on flow dynamics

The first method can provide very high detection accuracy in case of well-known open P2P protocols. It takes advantage in the investigation of some named P2P systems. Its drawbacks appear in high processor claim (for payload check), and the continuous change of P2P protocols, which are not available in most of the cases. Moreover, it also raises a number of legal and privacy problems. The second one is simpler to perform but it implies heuristic methods yielding less accurate results. However, it does not depend directly on actual P2P systems, thus it is more consistent and suitable for the analysis of P2P traffic aggregation. In this paper we have chosen the second approach and present an accurate and robust simple P2P traffic identification method.

A heuristic method for P2P traffic identification

Our proposed heuristic method consists of six steps, each being associated with a group of P2P flows to be identified. At the beginning we try to classify a set of widely used Internet applications (except P2Ps) based on well-known port analysis.

0. While port based analysis is less accurate to identify P2P traffic, it is still appropriate to distinguish traffic of common applications. Our search of these applications and their communication ports, in both TCP and UDP layers, results in a table of application ports (see Table 2). Flows with these ports in the *source_port* or *dest_port* are first extracted from the data sets. Web ports (80, 443, 8080, etc.) are not among these. The reason is that HTTP ports are not only used for web surfing but also by some P2P applications, e.g. KaZaA. The separation of web and P2P traffic is considered by the second heuristic.

Application	Port(s)	TCP/UDP
MSN Messenger	1863	TCP
Yahoo Messenger	5101, 5050	TCP
NETBIOS	135, 137, 139, 445	TCP and UDP
NTP	123	UDP
DNS	53	TCP and UDP
POP3	110	TCP
FTP	20, 21	TCP
…	…	…

Table 2. Some examples of common application ports

1. The first heuristic is based on the fact that many P2P protocols, e.g. eDonkey, Gnutella, Fasttrack, etc., use both TCP and UDP transport layers for communication. Reasonably the unreliable UDP is often used for control messaging, queries, and responses while data transmission relies on TCP. However, the large volume of UDP traffic observed in our measurement data indicates that UDP could also be used for data transfer. Thus by identifying those IP pairs which participate in concurrent TCP and UDP connections we can state that the traffic between these IP pairs is almost surely P2P. This heuristic is similar to what is proposed in [28] with a little difference. We note that some other common applications like NETBIOS, DNS also utilize both TCP and UDP. [28] needs a post-processing to extract this kind of traffic from the result of the heuristic. In contrast, this is not necessary in our case since we have already done this

in the initial (0th) step: these applications are among the common ones.

2. The second heuristic tries to separate web and P2P traffic from flows using HTTP/SHTTP ports, i.e. 80, 8080, 443, ... The typical difference between P2P and web communication of two hosts can be observed. In general, web servers use multiple parallel connections to hosts in order to transfer web pages text and images (also music, video contents in some cases). In contrast, data transmission between peers consists of one or more consecutive connections, i.e. only a single connection can be active at a time. This property is used to identify web servers, and then the traffic originating from them. The traffic using HTTP ports is divided into groups of individual IP pairs. The web server is the one with the IP address in the HTTP ports side which has parallel connections to its pair. We also differentiate between two cases: if the IP address of the web server belongs to the outside IP domain it is likely to be a public web server. Then all the HTTP traffic from them is marked as web traffic. In the other case only parallel flows with HTTP ports are marked as web traffic. The rest of this traffic group is P2P traffic. Unfortunately we realized that the most popular streaming applications (Windows Media Server, Helix Server, and Quick Time) can also use HTTP ports for transferring video or audio content. Since streaming data flows not necessarily have parallel connections to the web server, these data flows would mistakenly be identified as P2P flows. Although the amount of such streaming flows in the data sets seems to be small, we exclude the flows marked as P2P in this step from later analysis.
3. In the next step, P2P traffic is selected using default ports of P2P applications. P2P software often defines default ports for communication. It is true that in most cases peer users can change it to any arbitrary port (but it is not frequent since peer-to-peering is usually not prohibited for home users) or port can be dynamically chosen automatically or when firewall or port-blocking is observed. This step cannot detect all P2P connections, but once the traffic is collected we can be almost sure that it is from those concerned P2P systems. A table of well-known ports used by some popular P2P applications is collected for this step (see Table 3 for details). Flows containing these values in *source_port* or *dest_port* are all marked P2P.

P2P applications	TCP/UDP ports
Edonkey (eMule, xMule)	TCP 2323, 3306, 4242, 4500, 4501, TCP 4661-4674, 4677, 4678, 7778
FastTrack (*older* KaZaA)	TCP 1214, 1215, 1331, 1337, 1683, 4329
BitTorrent	TCP 6881-6889
Gnutella	TCP 6346, 6347
MP2P	TCP 41170, 10240-20480, 22321
DirectConnect (DC++,BCDC++)	TCP 411, 412, 1364-1383, 4702, 4703, 4662
ShareShare	TCP 6399, UDP 6388, 6733, 6777
Freenet	TCP 19114, 8081
Napster (File Navigator, WinMX)	TCP 5555, 6666, 6677, 6688, 6699-6701, 6257
SoulSeek	TCP 2234, 5534
Blubster	TCP 41170

Table 3. Network ports used by some popular P2P systems

4. In normal TCP/UDP operation, at least one of the two ports is selected arbitrarily. It is not likely that flows with similar flow identities (*source_IP, dest_IP, source_port, dest_port, prot_byte, TOS*) exist in relatively short measurements. This happens, however, in the case of P2P connections, if both source and destination peers dedicate a fixed port for data transfer. File download of a file is often executed in several smaller chunks. Therefore multiple flows with the same flow identities can be generated by P2P software. This is the basis of this heuristic: those identical flows are from P2P applications if at least two of each are found.
5. For the same reason as the above heuristic, it is not probable that a host (IP) will repeatedly choose a given arbitrary port for TCP/UDP connections unless it is a server. Web servers and other common server traffic is extracted by the previous heuristics, thus it is safe to introduce the next heuristic: if an IP uses a TCP/UDP port more than 5 times in the measurement period that {IP,port} pair indicates P2P traffic. The selected upper threshold (5) is a rule of thumb established empirically.
6. The last heuristic is based on the fact that objects of P2P downloads often have large sizes from several MB in case of music files or smaller applications to hundreds of MB in case of video files and larger software packages. In addition, peer users are patient. P2P downloads can last some ten minutes or hours. By this heuristic those flows are considered P2P flows which have flow size larger than 1 MB or flow length is longer than 10 minutes.

IV. Verification of the identification method

In order to examine the robustness of the heuristics presented in Section III a validation measurement was carried out. In this measurement besides gathering general and aggregated information of the traffic flows we also recorded the name of the corresponding application. This enabled us to validate the correctness of the proposed P2P traffic identification method.

The measurement program was written in C and used the *pcap* library to capture the incoming packets. Upon arrival of a new packet the program first determined which flow it belonged to, then updated the flow information, namely number of packets, number of bytes and end-timestamp. Both TCP and UDP flows have been identified by their source and destination IPs and ports. In addition, in case of TCP the SYN and FIN flags were also used to separate flows, while for UDP we used a timeout of 5 minutes.

The measurement collected the traffic generated by two Linux PCs running SMTP and web servers (although with very light traffic), and some P2P applications:

qtorrent, valknut, and aMule. These are the Linux clients of the Bittorrent, Direct Connect and eDonkey systems, respectively. To challenge the identification method, default ports of the P2P clients were modified. Several downloads have been initiated, meanwhile the P2P clients were also enable to serve requests of other peers. The measured trace contains more than 120000 data flows.

Heuristic step	Hit rate (%)
Known Applications	93.01
Heuristic 1	99.91
Heuristic 2 (HTTP ident.)	95.35
Heuristic 3	99.79
Heuristic 4	99.97
Heuristic 5	99.51
Aggregate P2P	**99.14**
Aggregate non-P2P	**97.19**

Table 4. Validation result of the identification method

We present the performance of each heuristic and the overall identification process in Table 4. The hit rate of each heuristics, counted in percentage, is the ratio of the number of *correctly marked* flows and the total number of *marked* flows by the heuristics. We note that the hit rate of the 6th heuristics is not shown in the table because it marked no flows in this data set. The last two rows in the table show the rate of correctly marked P2P (non-P2P) flows and the total number of P2P (non-P2P) flows in the data set. The result is very convincing in every statistics. The average hit rate is greater than 99.7%. The amount of unidentified traffic is about 0.1%. The ratio of wrongly marked P2P flows and unidentified P2P flows per the total marked P2P flows are 0.3% and 0.8%, respectively.

Note that these performance parameters are counted flow-wise. Similar results concerning the traffic quantities (bytes) are much better.

The validation of the proposed identification method is done for a relatively small measured traffic trace in which case the possible sources of errors are multiplied (expectable behavior used in some heuristics is more likely in large traffic measurements). Moreover, the modification of some P2P application ports made the task more difficult. Nevertheless the obtained result yields very convincing identification accuracy.

V. Identification results

As described earlier the traces are the sets of flow information collected using Cisco NetFlow measurements (see Section II). We assign a flag to each flow record of our database. The flag has the default value of *u* which means unknown (traffic) and it can be changed in the course of the identification process. The list of possible values of the flag is the following:

- *u*: unknown traffic (flow), default value
- *m*: management flow
- *o*: other non-TCP/UDP flow
- *k*: known common application (except HTTPs)
- *kh*: flow using Web ports (80, 443, 8080, etc.)
- *pX*: P2P flow, *X* denotes the heuristic which identifies the flow

First, flows used for management of the routers of traffic measurement are flagged as m. The management traffic is clearly identifiable by the IP addresses of the routers. Next flows of other IP protocols, which are not TCP and UDP, are marked by flag o. This type of traffic may consist of ICMP, IPv6, RSVP, GRE, IPsec ESP, etc. This step is obtained using the protocol byte information of the flow record. After that the proposed initial phase and the heuristic methods are applied to identify the common application traffic and the set of P2P traffic. The flags of unknown flows keep their default value u. The result of the identification procedure is summarized in Table 5. Note, that *Callrecords 1* and *2* data sets did not contain management traffic; non TCP/UDP traffic was also filtered out in a preliminary step.

Flag	# of flows (%)	Volume (%)
Data set	Callrecords 1 inbound	
k, kh	63.40	37.71
pX	35.35	62.19
u	1.25	0.11
Data set	Callrecords 1 outbound	
k, kh	71.83	15.62
pX	27.63	83.24
u	0.54	1.14
Data set	Callrecords 2 outbound	
k, kh	54.85	21.77
pX	44.62	77.50
u	0.53	0.73

Table 5. Traffic identification results

VI. Traffic analysis

In this study the analysis framework focuses on the fundamental differences between the P2P traffic and other Internet traffic (this will be referred to as *non-P2P traffic*). The comparison is done regarding several aspects of the traffic characterization.

Remember, that traffic flows, marked as P2P ‘*p2*’ by the second step of the heuristics, are excluded from the analysis (see Section III).

Overview of the traffic

The daily fluctuation of the traffic is presented in Fig. 2. The upper plots show the total and non-P2P traffic intensities of the *Callrecords 1* data set (including inbound and outbound direction), while the lower one shows the intensity and the flow count of the P2P traffic of the same set.

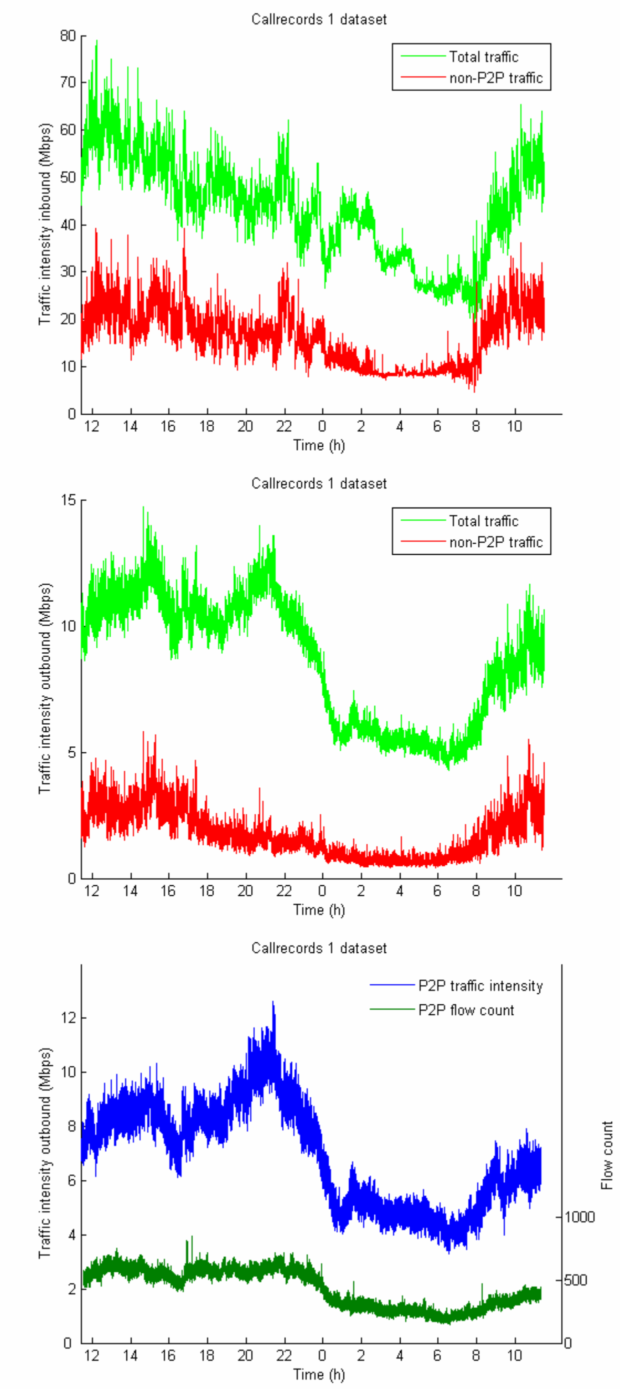

Fig. 2. Traffic intensities from *Callrecords 1* dataset

As observed in general, daily traffic can be divided into two parts: the busy period from around 8h to 24h and the non-busy period from about 0h to 8h. Both P2P and non-P2P traffic follow this daily tendency. In the case of non-P2P applications the traffic level shift between busy and non-busy periods is about ½ (the bandwidth falls to very low values in non-busy period), while in case of P2P applications the decrease of traffic intensity is somewhat smaller.

This is reasonable since non-P2P users, in general, do not generate traffic in the sleeping time. In contrast, P2P users (in our case also home users) turn on the P2P application and request some audio and video files (some can be very large). Then they leave the system to work over days, even when they are asleep during the night period. Basically, the P2P traffic can be steady over time, which can be seen in Fig. 2: the number of P2P flows has small variation (see the lower plot). We still see a certain decrease in the traffic. It happens since the number of downloadable sources decrease and probably more requests are not added during the night period.

The volume of P2P traffic, see also Table 5, which is about 60-80% of the total traffic, exceeds by far the traffic volume of the non-P2P applications. This observation is especially true for outbound aggregate traffic. The reason is that home users do not generate too much upload traffic, except for those users who use P2P applications. As a consequence the ratio of P2P traffic in the outbound direction is higher than in the inbound direction.

Number of P2P and total active users

In the measurement environment, Internet subscribes do not have fixed IP addresses. Each time a user connects to the Internet, a dynamic address is given to the user. Therefore it is impossible to determine exactly which data flow belongs to which user. However, less error is expected when we choose to associate an individual IP address to a user. Since the ADSL contracts at the present Internet provider do not limit the time of connections, the average connection time is relatively long. We assume that during our measurements, which lasted at most 24h, only a minimum number of IP address wanderings occurred.

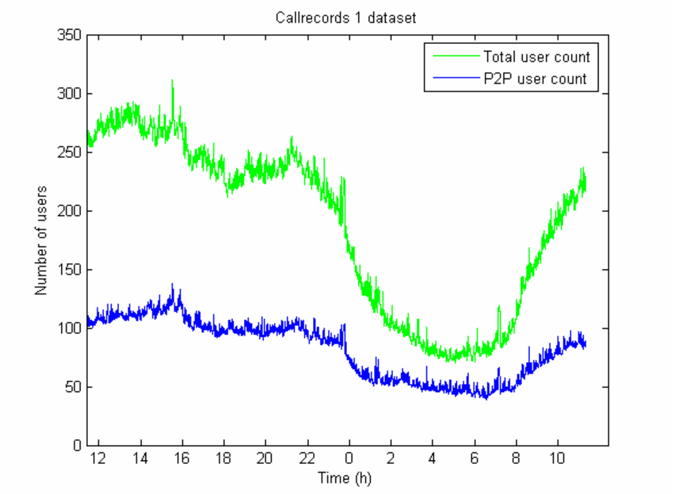

Fig. 3. The average number of P2P users (*Callrecords 1* dataset)

To calculate the number of active users, the number of different IP addresses participating in the flows is counted in every second. Then a sliding window of size 120s and step 50s is applied to smooth the variations caused by communication breaks. One of the results is shown in Fig. 3. The upper curve shows the fluctuation of the total number of users, while the lower curve displays the number of P2P users in the network. A user, who uses both P2P and non-P2P applications, is counted as P2P user. The total number of users, according to the time shift between busy and non-busy periods, decays as the non-busy period is approached. The lowest number of users is observed in the non-busy period. This similarity is not so striking in the case of P2P users. The answer is similar to the above; it is due to the typical behavior of P2P users/applications.

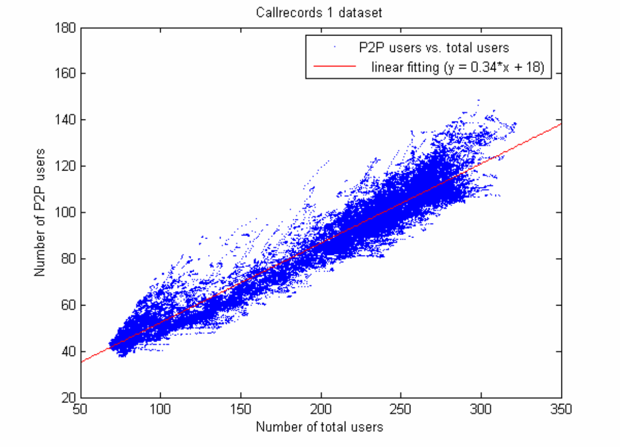

Fig. 4. Relation between P2P users and the total user number (*Callrecords 1* dataset)

The relation between the active P2P users and the total active users is presented in Fig. 4. As seen in the figure there is a strong linear connection between the two measures. This means that approximately a fixed ratio of active users is using P2P applications. This is quite an interesting finding and it is hard to find a reasonable explanation. However, if this relation is general, it would be very useful for e.g. traffic dimensioning. We plan to verify this relation in more different network environments. The estimated ratio between P2P users and total users is about 0.2 for this data set, 0.3 for the other two sets.

The relation between the number of active (P2P) users and the occupied bandwidth is also investigated. It is shown that a linear connection can be observed in both cases (P2P and non-P2P traffic). However, the variance of data around the assumed linear function is much higher than in the previous case (which is presented in Fig. 4). In addition, variation is higher and the slope of the line is much lower for non-P2P traffic. This means, P2P users (e.g. users, who use P2P applications as well) generate much more traffic in average than those users, who use only non-P2P applications.

Flow sizes and holding times

The next comparison is about the properties of data transferring: flow size and flow holding time. Fig. 6 presents the histogram of the flow sizes of P2P and non-P2P applications. We find no significant divergence in this characteristics. In both cases the plots, disregarding flow sizes smaller than 0.1 kB, nearly follow a straight line in the log-log scale. This indicates a possible heavy-tailed (Pareto) model for the flow size for both P2P (with shape parameter *a*=-0.3) and non-P2P flows (*a*=-0.25) and also for the overall traffic. (The assumptions of Pareto distribution were verified by several heavy-tailed tests: De Haan’s moment method, Hill estimator, and QQ-plot [34].) The number of P2P flows which are larger than about 100 kB is somewhat higher than the number of non-P2P ones, which is also reasonable, but the difference is not significant.

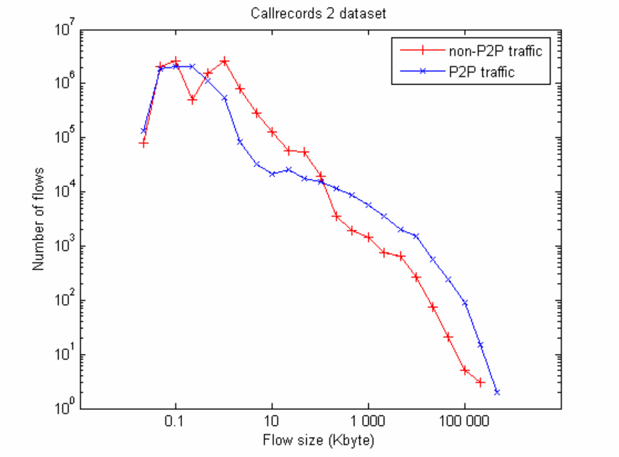

Fig. 5. Histogram of flow size (*Callrecords 2* dataset)

The result seems to be reconcilable with some newer developments of many P2P protocols. Independently of the size of the requested objects, at the beginning the P2P application downloads only a small chunk of the object. The condition of the network and source capacity is estimated from the characteristics of the previous downloads. The size of the next chunk will be determined according to the assumed download quality. Thus, at the end, the P2P traffic (concerning flow size in this case) behaves similarly as the non-P2P traffic.

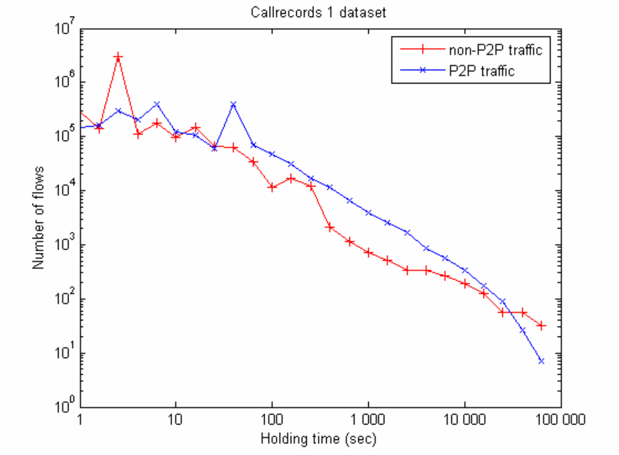

Fig. 6. Histogram of flow holding time (*Callrecords 1* dataset)

Similarity is also obtained in the flow holding time distribution of P2P and non-P2P traffic, see Fig. 7. Again, in the log-log scale, one can see two almost parallel lines in the two histograms. The plots suggest the Pareto distribution for both cases with the same shape parameter *a*=1.4. The shift in the histogram plot agrees with the fact that the total number of P2P flows is higher than that of the non-P2P ones by one order of magnitude.

Packet size distributions and typical packet sizes

We investigated the packet size distribution for packets belonging to P2P data flows and non-P2P data flows. The histograms of packet sizes for P2P and non-P2P traffic are shown in Fig. 7. Data packets, with size close to the MTU (Maximum Transfer Unit), appear frequently in the flows in both cases, and small packets about the minimum packet size are also common. Certain packet sizes have significant deviation from the average. However these small excursions do not make the packet size distribution of P2P and non-P2P traffic different. A clear trend is visible for both point-sets in Fig. 7.

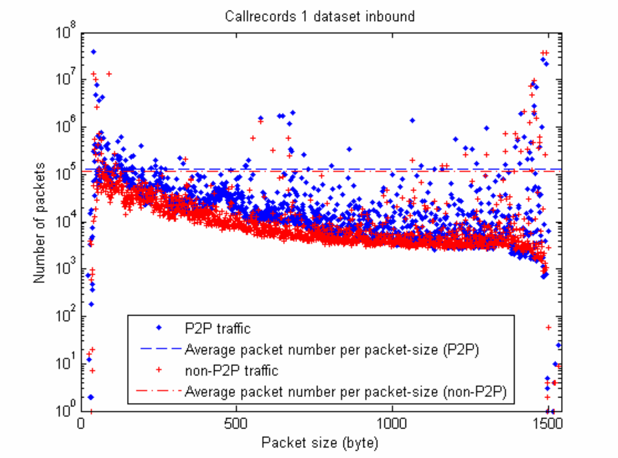

Fig. 7. Packet size distribution of P2P and non-P2P traffic

The huge deviation in certain packet sizes is due to certain applications. We tried to determine those applications, which are responsible for this deviation, causing certain packet sizes appear more frequent in the traffic flows. We calculated the average of the histogram values for both P2P and non-P2P traffic and plotted it with dashed line(s) in Fig. 7. This average value proved to be a good threshold, because it nicely separates frequent and infrequent packet sizes. We chose to investigate packet above this threshold. Remaining packet sizes, where the histogram value is under the threshold, were not investigated, because this way we could reduce the computational complexity of the subsequent steps.

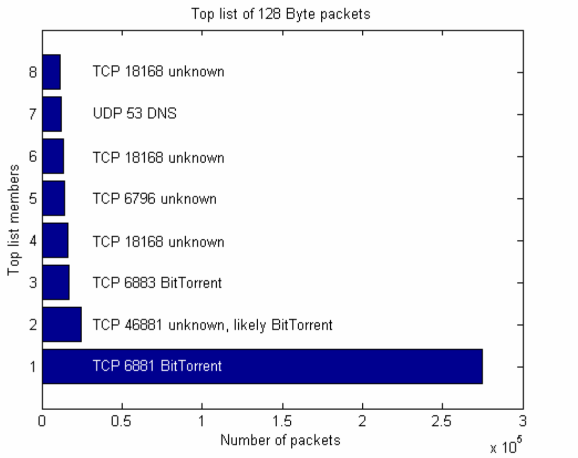

Fig. 8. Top list of source ports and applications for 128 byte data packets

After the filtering we calculated how data packets, belonging to a certain packet size, are distributed between source TCP (or UDP) ports. The “top list” of source ports was evaluated for every packet size, which were not filtered out in the previous step. The application, which generated the data packet, was identified based on the source port. Of course only applications with known communication ports could be surely identified. An example can be seen in Fig. 8 for 128 Byte data packets, suggesting 128 Byte data packets are typical for BitTorrent application.

A more detailed list of typical applications for different packet sizes is shown in Table 6. The “trustiness” measure means the ratio of packets with the given packet size belonging to the given application. Note, that an application for a certain packet size can be significant, even if the trustiness value is not so high (BitTorrent is significant for 128 byte packets, although the trustiness value is “only” 44%, see Fig. 8).

Packet size (byte)	Typical application	Trustiness (%)
89	SMTP	97.63
528	Gnutella	67.80
128	BitTorrent	44.86
86	eDonkey, BitTorrent	41.98
54	Remote Desktop	35.28
46	POP3	31.36
1200	eDonkey	29.97
58	eDonkey	24.85
64	eDonkey	21.64
120	eDonkey, DC++	21.03
74	eDonkey, BitTorrent	20.76
49	POP3	15.94
52	eDonkey	14.58
167	BitTorrent	14.25
50	MSN Messenger	9.74

Table 6. Typical applications for different packet sizes

Popularity distribution

The IP addresses were ranked according to their total amount of downloaded traffic. The downloaded traffic is plotted against the ranked IP address (which we have assumed to be associated with an individual user) in Fig. 9. The skewness in the popularity distribution of P2P systems is also justified in our analysis as in many studies of P2P traffic. The top 10% of P2P users are responsible for more than 90% of total download traffic. Our interest, however, is how it differs from the other Internet traffic. Our analysis shows that the difference does not lie at the head of the rank but at the tail. As we go down the rank, the download traffic by ranked users decreases very fast in the case of P2P users. There is a big split between “obsessive” and hobby P2P users. In contrast, the degree of traffic volume decay in case of ranked non-P2P users is very slow. The average non-P2P users create relatively stable traffic when they access the Internet: reading daily news, chatting with friends, etc.

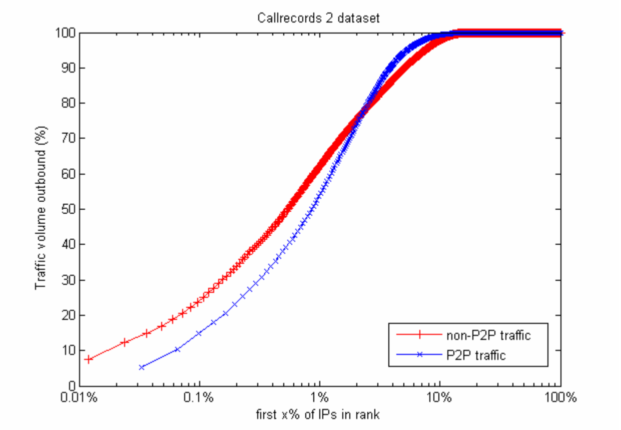

Fig. 9. Traffic volume of ranked IPs (*Callrecords 2* dataset)

At the top (about 10%) of the ranked list the popular Zipf's law seems to be accurate to describe both P2P and non-P2P traffic popularity. As seen in Fig. 10 two almost linear plot of P2P (marked by +) and non-P2P IP rank (marked by x) with an approximate slope of -1 indicates the standard Zipf distribution as the suitable model for *top ranked* users' traffic.

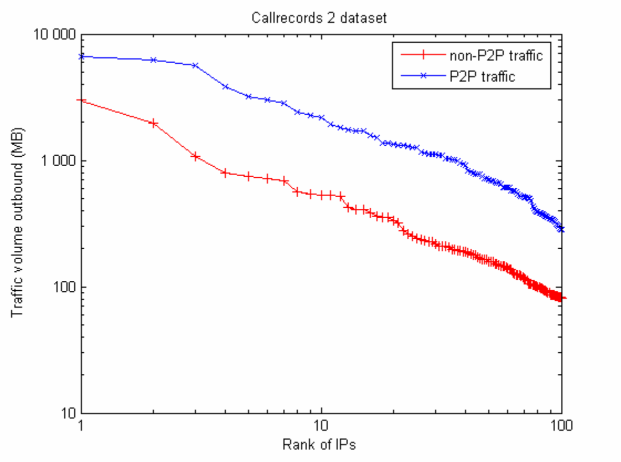

Fig. 10. Traffic vs. top ranked IPs (*Callrecords 2* dataset)

Analysis was also carried out for the connection population and similar curves were shown in the results. Fast decrease was observed in the case of P2P traffic as the ranking place increases, the decay is much lower in non-P2P case. In average a normal non-P2P user creates more, and probably smaller connections than P2P users despite that P2P traffic dominates in all measurements both in the volume and the connection number. This happens because, for example, opening of a web page involves multiple downloads of text, many images, and even audio and video elements.

Popular applications

We collected the most popular P2P and non P2P applications in order of transferred traffic. In the list of know applications HTTP is the absolute dominant. Note, that HTTP-port traffic can also include some streaming flows.

In the case of P2P only those P2P applications appear in the list, which have known (default) communication ports. From random or non-default communication port we cannot deduce the real application behind the port. Among P2P applications Direct Connect seems to be the most popular.

There is no strong correlation between the amount of transferred traffic and the number of traffic flows. We can observe for example, that the number of MSN Messenger flows is high; however the amount of transferred data is low.

Application	Traffic (MB)	# of flows
1. HTTP (including TCP 80 streaming)	127 449	2 208 968
2. FTP	30 744	7 459
3. POP3 + Secure POP3	3 891	117 820
4. HTTPs	1 578	46 333
5. Streaming (known port)	1 572	1 882
6. SMTP	1 338	65 473
7. SSH	267	1 293
8. MSN Messenger	221	24 345
9. IMAP	212	3 321

Table 7. Top list of known popular applications

P2P application	Traffic (MB)	# of flows
1. Direct Connect	6 184	87 368
2. Gnutella	4 746	151 357
3. BitTorrent	4 432	99 942
4. eDonkey	3 778	295 598
5. Napster, File navigator, WinMX	1 053	13 839

Table 8. Top list of known-port P2P applications

Discussion: the workload of P2P traffic aggregation

Our presented analysis may not be a complete comparative characterization of P2P and non-P2P traffic, but the attained analysis has highlighted some critical findings. P2P users/applications, by the typical content-sharing objectives of P2P usage, behave in a different way than other Internet applications. The difference manifests itself in the almost stable P2P activities over busy and non-busy time periods, the bandwidth-hungry nature, the skewness in the traffic volume distribution between P2P users, etc. However, the characteristics of P2P traffic aggregation, which would be a more important aspect from the service providers' and network operators' point of view, are quite similar to those of other traffic aggregation. While in the beginning P2P applications were confined to greedy file-sharing, nowadays they have grown up to be an inseparable component of the Internet due to several refined developments of P2P protocols. It has been shown that there is always a certain ratio of home users who use some P2P applications. The study establishes that the workload of P2P applications generates similar (heavy-tailed) flow size and flow holding time distribution like several non-P2P applications. As a consequence the P2P aggregation also shows the similar characteristics.

There may come the time when we should change the way of thinking about and treating P2P traffic. It is not an outstanding but an inseparable part of the overall Internet traffic just like every other traffic component.

VII. Conclusion

In this paper we first presented a novel P2P traffic identification method. The method collects a set of rules derived from the general behavior of P2P traffic. Our method does not use any payload information so it is easy to implement and use when payload cannot be evaluated because of legal or privacy obstacles or cannot be measured due to technical or financial problems. Our validation results show that the proposed algorithm is able to identify the P2P traffic very efficiently. The method was used to identify P2P traffic in current measurement data taken from one of the largest Internet providers in Hungary.

We also presented a comprehensive traffic analysis study focusing on the most important characteristics like the behavior of active users, the ratio between the P2P users and the total number of users, flow size and holding time distributions and the popularity distribution. We have found that the daily profile of P2P traffic intensity is less variable and shows a robust P2P user existence.

We showed that packet-level statistics of P2P and non-P2P data flows are basically similar. However there are some applications generating data packets with typical size. We investigated the relationship between packet sizes and applications resulting in a list of typical applications belonging to various packet sizes.

The analysis of the number of active users and total users revealed an almost linear relation. It suggests a very interesting and important result from a traffic dimensioning point of view: the ratio of active users and total users is almost constant (between 0.2 and 0.3 in our case). The study on flow sizes and holding times confirms earlier results showing the heavy-tailed behavior of both characteristics. We have also found that Zipf's law holds for P2P traffic popularity.

One of our major conclusions is that in spite of the different characteristics of individual P2P traffic the main characteristics of P2P aggregation (and this is important from the point of view of a service provider or a network operator!) do not differ significantly from the characteristics of other Internet traffic aggregation. Our future work will focus on the further investigation of this conjecture for general Internet traffic.

[1] E. Adar, B. A. Huberman, "Free Riding on Gnutella", Technical report, Xerox PARC, Aug. 2000.

[2] S. Saroiu, P. K. Gummadi, S. D. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems", in Proc. Multimedia Computing and Networking (MMCN'02), Jan. 2002.

[3] S. Saroiu, K. P. Gummadi, R. Dunn, S. D. Gribble, H. M. Levy, "An Analysis of Internet Content Delivery Systems", in Proc. 5th Symposium on Operating Systems Design and Implementation, Boston, MA, USA, Dec. 2002.

[4] 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. 19th ACM Symposium on Operating Systems Principles (SOSP-19), Bolton Landing, NY. Oct. 2003.

[5] T. Karagiannis, A. Broido, N. Brownlee, K. Claffy, M. Faloutsos, "File-Sharing in The Internet: A Characterization of P2P Traffic in The Backbone", Technical Report, UC Riverside, 2003.

[6] S. Sen, J. Wang, "Analyzing Peer-to-Peer Traffic Across Large Networks", *IEEE/ACM Transactions on Networking*, 12(2):219-232, 2004.

[7] K. Tutschku, "A Measurement-based Traffic Profile of the eDonkey Filesharing Service", PAM 2004: 12-21.

[8] J.A. Pouwelse, P. Garbacki, D.H.J. Epema, H.J. Sips, "The Bittorrent P2P File-Sharing System: Measurements And Analysis", 4th Int. workshop on Peer-to-Peer Systems (IPTPS'05), Feb. 2005.

[9] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose, D. Towsley, "Modeling Peer-Peer File Sharing Systems", in Proc. INFOCOM'03, San Francisco, CA, Mar. 2003.

[10] K. K. Ramachandran, B. Sikdar, "An Analytic Framework for Modeling Peer to Peer Networks", in Proc. INFOCOM'05, 2005.

[11] G. de Veciana, X. Yang, "Fairness, Incentives and Performance in Peer-to-Peer Petworks", in Proc. Allerton Conf. on Communication, Control and Computing, 2003.

[12] X. Yang, G. de Veciana, "Service Capacity of Peer to Peer Networks", in Proc. INFOCOM'04, 2004.

[13] D. Qiu, R. Srikant, "Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks", in Proc. ACM SIGCOMM'04, Portland, OR, Aug. 2004.

[14] B. Yang, S. Kamvar, H. Garcia-Molina, "Addressing the Non-Cooperation Problem in Competitive P2P Systems", Workshop on Peer-to-Peer and Economics, Jun. 2003.

[15] D. Hughes, I. Warren, G. Coulson, "Improving QoS for Peer-to-Peer Applications through Adaptation", in Proc. of the 10th Int. Workshop on Future Trends in Distributed Computing Systems (FTDCS 2004), Suzhou, China, May 26-28, 2004.

[16] E. Kalyvianaki, I. Pratt, "Building Adaptive Peer-To-Peer Systems", in Proc. 4th Int. Conf. on Peer-to-Peer Computing (P2P'04), 2004.

[17] M. Iguchi, M. Terada, K. Fujimura, "Managing Resource and Servent Reputation in P2P Networks", in Proc. 37th Annual Hawaii Int. Conf. on System Sciences (HICSS'04), 2004.

[18] G. Ding, B. Bhargava, "Peer-to-Peer File-Sharing over Mobile Ad hoc Networks", in Proc. PerCom Workshops, 2004.

[19] M. Demirbas, H. Ferhatosmanoglu, "Peer-to-Peer Spatial Queries in Sensor Networks", in 3rd IEEE Int. Conf. on Peer-to-Peer Computing (P2P'03), Linkoping, Sweden, Sept. 2003.

[20] M. Roussopoulos, M. Baker, D. S. H. Rosenthal, T. J. Giuli, P. Maniatis, J. C. Mogul, "2 P2P or Not 2 P2P?", IPTPS 2004: 33-43.

[21] Y. Guo, K. Suh, J. Kurose, D. Towsley, "A Peer-to-Peer On-Demand Streaming Service and Its Performance Evaluation", in Proc. IEEE Int. Conf. on Multimedia \& Expo (ICME 2003), Baltimore, MD, Jul. 2003.

[22] G. Cugola, G. P. Picco, "Peer-to-Peer for Collaborative Applications", Int. Workshop on Mobile Teamwork Support, Vienna, Austria, Jul. 2002.

[23] C. Buragohain, D. Agrawal, S. Suri, "A Game Theoretic Framework for Incentives in P2P Systems", in Proc. 3rd Int. Conf. on Peer-to-Peer Computing, 2003.

[24] J. Shneidman, DC Parkes, "Rationality and Self-Interest in Peer to Peer Networks", in Proc. 2nd Int. Workshop on Peer-to-Peer Systems (IPTPS'03), 2003.

[25] T. Risse, P. Knezevic, A. Wombacher, "P2P Evolution: From File-sharing to Decentralized Workflows", *it-Information Technology*, 4:193--199, Oldenbourg, 2004.

[26] S. Sen, O. Spatscheck, D. Wang, "Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures", in Proc. 13th Int. Conf. on World Wide Web, NY, USA, 2004.

[27] M. Kim, H. Kang, J. W. Hong, "Towards Peer-to-Peer Traffic Analysis Using Flows", DSOM 2003: 55-67.

[28] T. Karagiannis, A. Broido, M. Faloutsos, K. Claffy, "Transport Layer Identification of P2P Traffic", in Proc. 4th ACM SIGCOMM Conf. on Internet Measurement, Taormina, Sicily, Italy, Oct. 25-27, 2004.

[29] T. Karagiannis, A. Broido, N. Brownlee, K. Claffy, M. Faloutsos, "Is P2P dying or just hiding?", IEEE Globecom, Dallas, Texas, November 2004.

[30] Internet2 NetFlow: Weekly Reports - http://netflow.internet2.edu/weekly/

[31] A. Gerber, J. Houle, H. Nguyen, M. Roughan, S. Sen, "P2P The Gorilla in the Cable", in National Cable & Telecommunications Association (NCTA) 2003 National Show, Chicago, IL, June 8-11, 2003.

[32] R. Meent, A. Pras, "Assessing Unknown Network Traffic", CTIT Technical Report 04-11, University of Twente, Netherlands, February 2004.

[33] S. Ohzahata, Y. Hagiwara, M. Terada, K. Kawashima, "A Traffic Identification Method and Evaluations for a Pure P2P Application", Lecture Notes in Computer Science, p55 Vol. 3431, 2005.

[34] S. I. Resnick, "Heavy Tail Modeling and Teletraffic Data", *The Annals of Statistics*, 25(5):1805--1869, 1997.