Towards Open Tracing of P2P File Sharing Systems

Danny Hughes
Computing, InfoLab21,
Lancaster University,
Lancaster, UK.
+44 (0)1524 510352
danny@comp.lancs.ac.uk

Kevin Lee,
School of Computer Science,
University of Manchester,
Manchester, UK.
+44 (0) 161 2756132
klee@cs.man.ac.uk

James Walkerdine
Computing, InfoLab21,
Lancaster University,
Lancaster, UK.
+44 (0)1524 510352
walkerdi@comp.lancs.ac.uk

Abstract

Since the release of Napster in 1999, peer-to-peer file-sharing has enjoyed a dramatic rise in popularity. A 2000 study by Plonka on the University of Wisconsin campus network found that file-sharing accounted for a comparable volume of traffic to web applications, while a 2002 study by Saroiu et al. on the University of Washington campus network found that file-sharing accounted for more than treble the volume of web traffic observed, thus affirming the significance of P2P in the context of Internet traffic. Empirical studies of peer-to-peer traffic are essential for supporting the design of next-generation peer-to-peer systems, informing the provisioning of network infrastructure and underpinning the policing of peer-to-peer systems. This paper surveys existing work in the field of peer-to-peer monitoring and based upon this assessment of the state-of-the-art describes the design and implementation of the Open P2P tracing system. This system aims to improve the research community's understanding of P2P file sharing systems by providing continuous and up-to-date traffic data which is anonymized and made freely accessible to all interested parties. Data from this system has been used in a variety of projects and papers, which are used to illustrate the broad range of research that can benefit from an open tracing system.

Keywords: Peer-to-Peer (P2P), File sharing systems, Monitoring approaches

1. Introduction

Since the release of Napster [1] in 1999, peer-to-peer (P2P) file-sharing has enjoyed a meteoric rise in popularity, to the point that P2P applications are now widely considered to be responsible for more traffic than any other Internet application [2]. Given the scale of P2P traffic, understanding traffic characteristics is of critical importance and has specific benefits in the context of: i) provisioning network infrastructure, ii) informing network policy, iii) informing the design of new P2P applications, iv) managing existing P2P communities and, v) policing P2P systems.

Many significant studies of P2P file sharing systems have been performed. These studies have illuminated a range of P2P characteristics; however, we believe that there remain significant shortcomings in the current body of research on P2P file sharing systems. These shortcomings include:

- ***The extensive use of closed data sets***, which prevents the findings of existing studies being revisited. Furthermore, as truly representative P2P traces may take months or even years to perform, the use of closed data-sets has led to significant and unnecessary duplication of effort.
- ***Trend analysis*** is poorly supported by existing studies, which, with a few exceptions [3] [4], are not of sufficient duration to reveal long-term trends in user behaviour.
- ***Cross discipline perspectives*** are often lacking in existing studies, which tend to concern themselves largely with technical factors and often fail to consider factors such as group psychology, the economics of file sharing and the ethics of monitoring real-world distributed systems.

We suggest that these shortcomings may be addressed through the development of an 'Open P2P Tracing System' [37] which aims to produce a significant, public and freely accessible data-set. Such a system would monitor P2P traffic on a long-term basis and make it available in near real-time, allowing the identification of trends and the revisiting of data points by researchers using different methodologies. Access to the data should also be simplified as far as possible to encourage the use of this data set by researchers from non-computing backgrounds and in particular sociology, psychology, economics and law.

The remainder of this paper is structured as follows: Section 2 provides a brief classification of P2P monitoring methodologies. Section 3 surveys the state-of-the-art in P2P monitoring technologies and studies. Section 4 discusses the limitations of current P2P approaches. Section 5 presents the design of an Open P2P Tracing System. Section 6 describes the initial evaluation of this system. Section 7 discusses how the open tracing system has been exploited to perform research. Finally section 8 discusses avenues for future work and concludes.

2. P2P Tracing Methodologies

Empirical studies of P2P systems may be classified as using one of three broad tracing methodologies: network-level tracing, passive application-level tracing or active application-level tracing [5].

Network-level traces are performed by deploying code on core or gateway network infrastructure and performing IP-level packet monitoring. Network-level tracing is transparent to the P2P network, however, this approach introduces local bias,

which results from deployment location and accurate identification of P2P traffic can be highly problematic.

Passive application-level traces are performed by monitoring the messages passed at the application level. In modern decentralized file-sharing systems all peers participate in message passing and therefore passive monitoring can be achieved simply by modifying a peer to log the messages that it is required to route. Passive application-level tracing is transparent and may be performed without access to core network infrastructure, though the rate at which data can be gathered using this methodology is significantly lower than that of network-level tracing.

Active application-level traces address the scalability shortcomings of passive application-level tracing by employing an aggressive querying and connection policy wherein the monitoring peer attempts to reconnect to and interrogate as much of the application-level network as possible; 'crawling' the P2P network in order to maximize the size and typicality of trace data. While this approach improves the quality of trace-data and the speed at which it is acquired, it does so at the expense of transparency due to the disruptive effect of repeated reconnections and high message generation on the P2P system being monitored.

Section 3 discusses significant empirical studies of P2P file sharing networks, organized according to the tracing methodology used. The findings of these studies are summarized along with their shortcomings.

3. Empirical Studies of P2P File Sharing systems

This section presents significant P2P traffic monitoring studies belonging to each of the tracing methodology classes introduced in section 2, spanning the period from 2000 to 2008. The specific methodology of each study is described alongside its significant findings. Based upon this survey, the benefits and limitations of each class of monitoring approach are discussed along with the general limitations of current P2P studies. While it is impossible to perform an exhaustive study in a single paper, this survey covers the most significant and oft cited studies of P2P networks.

3.1. Network-Level Monitoring

The first network-level study of P2P traffic was performed by Plonka et al. [6]. This study analyzed the bandwidth consumed by Napster [1] on the University of Wisconsin-Madison network during March 2000. A seven hour trace was gathered using a specially developed tool called FlowScan to monitor Napster traffic. FlowScan first identified nodes communicating with the napster.com servers as potential P2P participants and then applied simple heuristics to the node's incoming and outgoing traffic in order to identify Napster-related traffic. The Plonka study found that as early as 2000, P2P applications generated a comparable volume of traffic to the web at 23.1% of total bandwidth, compared to 20.9% for web traffic. Unfortunately, it is difficult to assess the accuracy of this study due to the lack of published details regarding the FlowScan traffic-catagorisation system. However, the short duration of the trace is likely to have resulted in inaccuracy, particularly as other studies have found significant time-of-day variations [10]. Nevertheless, the Plonka study was useful in highlighting the increasing bandwidth consumption observed during the early days of P2P applications.

Plonka's observations on the growing volume of traffic being generated by P2P applications were corroborated by in June 2002 by a University of Washington study conducted by Saroiu et al [2] Their nine day trace found that P2P traffic consumed 43% of campus bandwidth, compared to just 14% for web traffic - a significant increase since the Plonka study. The Saroiu study identified traffic generated by the two dominant P2P systems of the day; Gnutella 0.4 [22] and Kazaa [13] based upon common port usage. In addition to raw traffic data, the Saroiu study reported more fine-grained information about the P2P work-load. This included the finding that, on average, objects retrieved from P2P networks were three orders of magnitude larger than objects retrieved from the web along with the finding that a small subset of peers are responsible for the majority of P2P traffic - a finding that corroborates the results obtained by Adar et al [7] in their passive application-level study (see section 3.2).

Gummadi et al. continued P2P monitoring work at the University of Washington with a 200-day trace of Kazaa traffic in 2003 [3]. This was recorded using a similar methodology to the 2002 trace, except that traffic was identified based upon Kazaa-specific HTTP headers rather than by port use. Uniquely Gummadi's 2003 trace was long enough to observe seasonal variations in P2P traffic and the effect of changing network policies on P2P workloads. Using this trace, Gummadi developed a detailed parameterized model of P2P workloads, which can be used by developers to generate realistic evaluation data.

Accurate identification of P2P traffic is a vital component of network-level P2P monitoring. In the case of the Plonka trace [6], identification was simplified by Napster's semi-centralized architecture [1], while the Saroiu [10] and Gummadi [3] trace identified traffic by port number and header data respectively. However, recent research [20] has demonstrated that users are increasingly moving to P2P systems that aim to avoid monitoring through the use of non-standard ports and encrypted header data. To address this issue, Subhabrata et al. [23] have developed a system for real-time network-level identification of P2P traffic. This system was implemented as an extension to the AT&T's Gigascope [24] high speed traffic monitor. Subhabrata et al. evaluated their traffic identification approach using a 24 hour week-day trace and an 18 hour weekend trace gathered in November 2003 on a major internet backbone. This was augmented with a 6 day trace of traffic on a VPN where network administrators attempt to block P2P traffic, also conducted in November 2003. Subhabrata's approach proved capable of identifying traffic from today's popular P2P systems in real-time for traffic flows of up to 1Gbps while maintaining misidentification rates of less than 5%. While the trace data gathered for this study was used to evaluate their traffic monitoring approach, the authors did not attempt to further characterize or examine the P2P traffic that they observed. The extended version of Gigascope used in this

study is capable of identifying traffic from Gnutella [12], Fasttrack [4], eDonkey [13], Direct Connect [14] and Bittorrent [16]. Subhabrata's identification approach is based upon the flexible concept of *application signatures*, which can be used to categorize traffic using a wide range of metrics.

The growing use of public and application independent anonymization services such as Tor [25] provides an interesting new target for network-level monitoring. Tor is itself a P2P system, which uses an overlay network of routers to enable anonymous outgoing and incoming connections. Any traffic sent via Tor is forwarded from peer to peer on the Tor overlay, ultimately reaching an exit peer, at which point the packet is forwarded on to its original destination. Viewed from the destination, the traffic appears to originate at the Tor exit node, thus protecting client user's identity. Tor also allows nodes participating in the overlay to provide services which may be accessed as though they are hosted at the exit node, allowing for the anonymous hosting of internet services. As Tor is effectively application independent, it is possible to implement network-level monitoring by hosting an exit peer and monitoring the outbound and inbound traffic. Standard network-level monitoring tools may be used for this purpose just as they would on a network gateway; however, monitoring Tor traffic has two significant advantages over monitoring at gateway locations. Firstly, as Tor is accessible world-wide, monitoring Tor exit peers are unlikely to exhibit geographic bias. Secondly, no special access to network infrastructure is required.

The first study of Tor was performed by Bauer et al. at the university of Colorado, Boulder in 2007 [26]. Bauer analyzed Tor's vulnerability to attack and discovered an inverse relationship between the degree of optimization in the Tor overlay and its resilience against attack. In 2008, McCoy et al. extended Boulder's work by performing a detailed characterization of Tor traffic [27] using the methodology outlined above. The study first provided a breakdown of Tor traffic by type, finding that web and Bittorrent data makes up the majority of traffic. The study also analyzed the location of Tor users, finding a significant geographic bias. McCoy also found that a significant level of Tor misuse occurs, for example snooping on plain-text data such as POP email traffic. Loesing et al. performed a detailed performance measurement of Tor traffic in 2008, analyzing the QoS properties and specifically the latency of Tor. Based upon this detailed analysis, performance optimisations to the Tor protocol were suggested [35].

In each of the studies discussed above, network-level tracing was used to record the low-level characteristics of P2P traffic flows on private networks. Network-level tracing is potentially transparent, scalable and allows comparison of traffic from multiple domains side-by-side. However, with the exception of Tor monitoring, this approach is dependent upon access to core network infrastructure, which is not always feasible. While researchers may have access to gateway infrastructure on large private networks, such as academic networks, data obtained from such sources should be viewed as potentially biased due to differences between the characteristics of the private network's users and general Internet users.

3.2. Passive Application-Level Monitoring

The first passive application-level trace of a P2P system was performed by Adar and Huberman in 2000 on the Gnutella 0.4 network [7]. This 24-hour trace logged resource-discovery traffic which was then used to assess the prevalence and characteristics of a problem known as 'free riding', wherein users download resources from, but do not upload resources to a P2P file-sharing system. The Adar trace was performed by modifying the open-source 'Furi' Gnutella client (no longer available) to monitor search, response and peer discovery messages. Adar and Huberman discovered that participation in Gnutella was highly asymmetrical with 66% of peers sharing no files at all and almost 50% of all files being served by the top 1% of hosts. This finding was significant as it contradicted the (then) conventional wisdom that user participation in P2P file sharing systems is symmetrical. Adar's result was later corroborated by Saroiu's 2002 network-level study [10].

Hughes et al [4] revisited the results of the Adar trace in 2004 on the Gnutella 0.6 network [12] based upon a one week trace. The trace was performed using a specially developed monitoring tool based on the Jtella base classes [17]. The monitoring peer connected to the Gnutella network as an Ultrapeer [12] and periodically reconnected in order to maximize the size and typicality of its sample-base. Hughes discovered that in the four years since the Adar study, the proportion of free-riders had increased from 66% to 85%, while corroborating Adar's finding that the top 1% of hosts serve almost 50% of all files. Hughes speculated that the increase in free riding may be the result of an increase in prosecution of copyright infringement. Hughes et al. revisited this data point using their 2005 trace and found that the level of free riding on Gnutella had continued to increase - to over 95% [28]. This trace was later used to assess the level of illegal pornographic material being distributed on the Gnutella network [8]. The study found that an average of 1.6% of searches and 2.4% of responses contained references to illegal pornography, though this material is distributed by a tiny subset of peers that typically share nothing else. This result was subsequently refined in 2008, looking only at the level of traffic relating to child abuse media. This study found that more than 1% of search traffic and 1.6% of search-response traffic relates to child abuse related media.

In each of the cases discussed above, passive application-level monitoring is used to study application level properties in an Internet-wide context. Like network-level monitoring, passive application-level monitoring is transparent, however, it does not require access to low-level network infrastructure. Unfortunately, in cases where a very large sample of network traffic is required quickly, passive monitoring would be unsuitable due to the small-world properties of modern P2P networks.

3.3. Active Application-Level Monitoring

Ripneau and Foster [9] performed the first active application-level trace of the Gnutella network from November 2000 to May 2001. This study attempted to map the Gnutella network in terms of the average number of links between hosts and the number of hops that these links represent on the underlying IP network. To achieve this, a specialized Gnutella peer known as

a '*crawler*' was developed. The crawler connects via the normal Gnutella boot-strapping system and uses Gnutella's peer-discovery mechanism [12] to find new peers. The IP address of these peers is added to the list of those observed and the crawler attempts reconnection in a new location, repeating the process and gradually building a 'map' of the network. The resulting map includes the total number of nodes, the total number of links and average traffic data. Based upon the findings of this study, Ripneau concluded that the emergent structure of the Gnutella network was such that the network's bandwidth consumption would limit its scalability, as predicted by Ritter [29]. Unfortunately, Ripneau's crawling approach is invasive, as repeated reconnection affects the P2P network. It is also un-scalable due to the computational and network expense incurred when crawling the application-level network.

Saroui et al. [10] extended Washington University's work on monitoring P2P systems to the application level with a one month crawl of the Gnutella network in May 2001. The crawler used a similar methodology to Ripneau and observed between 8,000 and 10,000 unique peers, which at that time would have accounted for between 25% and 50% of the Gnutella network. The 2001 Saroiu trace recorded low-level data, including each peer's IP address, latency and bottleneck bandwidth between peers; along with higher level data including each peers advertised bandwidth and the number and size of files being shared. These high-level properties were measured by logging Gnutella's resource discovery and network maintenance messages, while bottleneck bandwidth was measured using SProbe [30], a network tool that uses a TCP exploit to accurately measure bottleneck bandwidth without the need for remote cooperation.

Chu et al [11] performed the first study that attempted to quantify the availability of peers and files on the Gnutella network using a forty day trace performed in early 2002. This trace was gathered by a tool based upon the Jtella API [17] that followed a similar methodology to the Ripneau crawler [9]. Search-response messages were intercepted by the crawler and unique peers were identified based upon their advertised IP and port pairs. The crawler was used to gather a list of 20,000 unique peers using the 'BearShare' and 'SwapNut' clients, at which point a second program, known as the 'tracking manager' attempted to download each peer's file-list using proprietary BearShare and SwapNut extensions. Using this methodology, the availability of peers and files was monitored for a period of 40 days beginning on March 28th. Chu reported a strong correlation between time-of-day and node availability and proposed a model to describe peer availability. Additionally, Chu provided a breakdown of relative file-type popularity and corroborated the finding of Saroiu [10], that file popularity is highly skewed with the top 10% of files accounting for more than 50% of shared data. A clear limitation of Chu's study lies in the use of proprietary extensions to obtain file lists, which limits the size of the trace and introduces possible bias due to the limited user-group studied.

Bittorrent is peculiar amongst P2P file sharing systems in that it does not implement a resource discovery mechanism. Thus, passive application-level monitoring is rendered ineffective due to the lack of resource discovery traffic. For this reason, Bittorrent studies generally actively query Bittorrent 'trackers'; specialized peers which mediate connection to the file distribution overlay, joining this overlay to participate in file distribution if further data is required. The first Bittorrent study was performed by Izal et al. in 2004 [31] shortly after Bittorrent's release and analyzed the life-span of a files being distributed using Bittorrent over a five month period. Izal showed that Bittorrent was a highly effective distribution mechanism for popular media effectively addressing the problem of 'flash crowds'. In 2005 Thommes et al. [32] further examined the 'fairness' of the Bittorrent protocol. The study found that Bittorrent performs near-optimally in terms of uplink bandwidth utilization, but that low bandwidth peers frequently downloaded significantly more than they uploaded. While the differential experience of Bittorrent peers may be considered unfairness, as it was by the authors of this paper, it may also be responsible for Bittorrent's high level of popularity; as the protocol is capable of catering to the needs of users on both high bandwidth and slow connections.

In each of the cases discussed above, active application-level monitoring has been used to study P2P traffic properties in an Internet-wide context, where a very large and typical body of trace data was required (e.g. mapping the Gnutella network). Active application-level monitoring is easy to deploy and should not contain local bias; however, the aggressive reconnection and interrogation approach employed makes this approach invasive and limits its scalability. Due to the invasiveness of this approach, active application-level monitoring may be easily detected and due in part to extensive copyright enforcement activities, file sharing user communities actively search for and attempt to circumvent active monitoring approaches.

3.4. Summary of Monitoring Approaches

This paper introduced a classification scheme for empirical studies of P2P file sharing systems based upon the tracing methodology that they employ: network-level monitoring, passive application-level monitoring or active application-level monitoring. In the context of this classification, significant empirical studies were reviewed along with the benefits and drawbacks of each approach. These are summarized below:

Network-level monitoring is transparent to the network and highly scalable. It is capable of comparing traffic flows from multiple P2P systems side-by-side and is well suited to characterizing P2P traffic on large private networks; however, it is poorly suited for performing global monitoring of P2P systems due to the possibility of local bias. Moreover, network-level monitoring requires low-level access to core network infrastructure, which is often unfeasible. Examples of network-level monitoring studies include [2] and [6].

Passive application-level monitoring is also scalable and transparent to the network. It can be performed without access to core network infrastructure, though it does not provide as large a volume of trace data as network-level monitoring or crawler-based application-level monitoring. Furthermore, it is inherently protocol specific. Passive application-level monitoring is thus best suited to instances where network-level monitoring is impossible or where a non-invasive approach is

desirable. Examples of passive application-level monitoring studies include [7] and [8].

Active application-level monitoring is less transparent and scalable than either network-level or passive application-level monitoring; however, it allows large volumes of trace data to be gathered without low-level access to the network infrastructure. It is thus the best approach where global network information is required and access to the underlying network infrastructure is not possible. Examples of active application-level monitoring studies include [10] and [11].

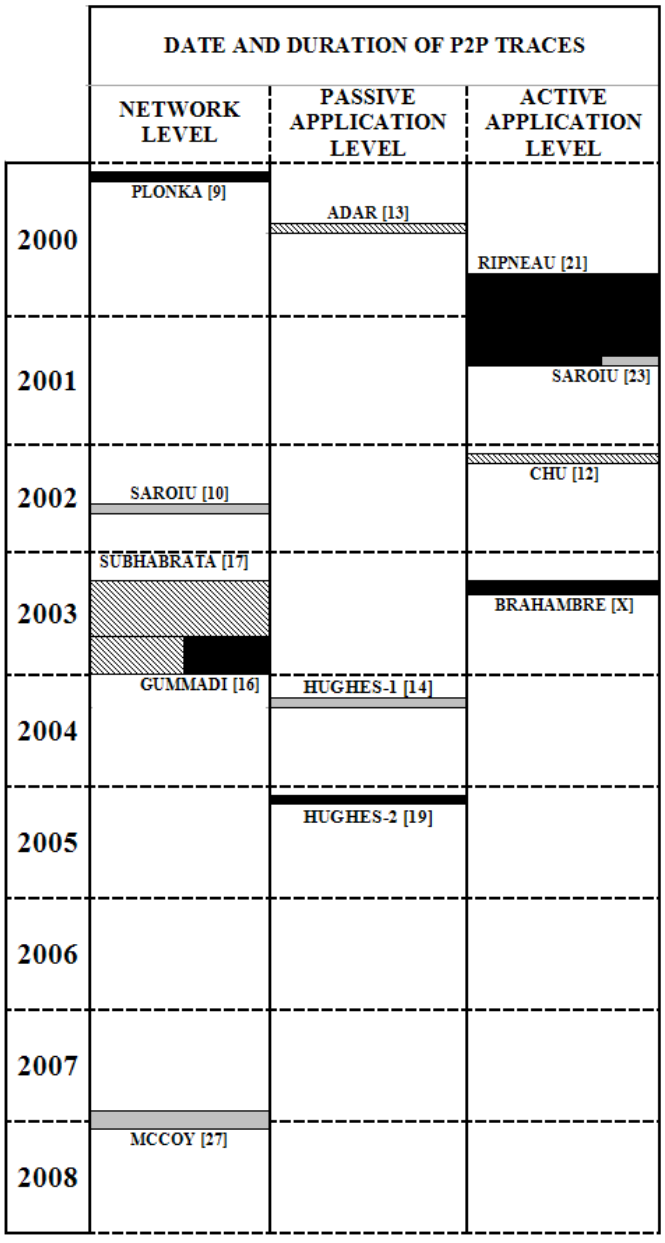

Figure 1. Time Distribution of P2P Traces

P2P traces such as those presented in this paper have proven invaluable in informing research in the field of P2P systems, however, each of these studies provides only a piece of the puzzle; describing a subset of P2P traffic characteristics for a subset of protocols over the duration of the trace. Often, papers which cite these studies fail to adequately consider such limitations. For example, the data-point provided by Adar's 2000 study of free riding [7] has been used in a significant body of research until the present day, however, when this study was revisited by Hughes et al. [4] in 2005, it was discovered that free riding had increased, revealing a significant, and (until that point), unidentified trend.

Figure 1 illustrates the date and duration of each of the P2P traces discussed in this paper. As figure 1 illustrates, few of the P2P studies presented in this paper are of sufficient duration to identify trends in P2P traffic, rather they simply provide a data-point for the monitored characteristics. The notable exceptions to this are Gummadi's 2003 Kazaa trace [3] which was long enough to observe seasonal variations and Hughes' 2005 study of free-riding [5] which, by revisiting Adar's 2000 experiment [7] was able to show an intervening trend in user behavior.

4. Limitations of Existing Work

There are a number of significant shortcomings in the current body of research on P2P traffic monitoring. The first and perhaps most significant of which is the wide-spread use of closed data sets. As can be seen from Figure 1, P2P studies may require weeks or even months of P2P traffic data. While it is understandable that after investing significant time and effort in gathering a data set, researchers may be reluctant to make this data public, this prevents the findings of studies being verified using different methodologies and prevents trace data being revisited in new contexts.

Another significant gap exists in the body of work on P2P monitoring regarding the identification of underlying trends. For example, the data-point provided by Adar's 2000 study of free riding [7] was revisited by Hughes in 2004 [5] and 2005 [28], each time revealing a significantly different data point. It may be is possible that other equally significant trends might be discovered by revisiting past studies. For example, would the growing popularity of digital video be reflected by an increase in the availability of such files since Chu's [11] 2002 study of file availability? Despite the possibility of exposing significant trends in user behaviour, few studies choose to revisit earlier data-points.

Most empirical studies of P2P file sharing systems are concerned only with the technical characteristics of P2P traffic (files shared, bandwidth usage etc.). While this information is critical for simulation of P2P traffic and for the development of approaches to encouraging positive user behaviour, the next step, reasoning about the social and psychological factors which produce this behaviour, is rarely taken. Furthermore, most studies do not take into account the real-world factors which may affect P2P traffic. Notable exceptions to this are the studies by Adar [7] and Hughes [5] [8] [28] [33], that explicitly consider the social factors which are responsible for observed behaviour.

5. Design of an Open P2P Tracing System

This paper has made the case that an open, easy to access and long-term P2P trace is required to improve our understanding of P2P file sharing systems. This section now discusses the design and implementation of such a system: The Open P2P Tracing System. As previously described, the system will use a passive application-level tracing methodology [5] to gather data. The implementation of this functionality will now be described.

5.1. Tracing Functionality

Implementation of tracing functionality is dependent upon the P2P system being monitored. As the Open Tracing System aims to provide a widely reusable data set, we intend to monitor several of today's most popular P2P systems, including Gnutella [12], Fasttrack [13], eDonkey [14], DirectConnect [15] and Bittorrent [16]. In order to minimize the time required to port monitoring code to additional P2P networks we implement logging functionality by modifying existing open source clients available for each P2P network. Analysis of such clients, which include Jtella [17], Open DirectConnect [18] and Azureus [19] revealed that each shared elements of common structure. Of particular significance in terms of implementing tracing support was that each client implements a single routing component which is used to process incoming and outgoing messages. It is into this routing component that we insert monitoring code. This is shown in Figure 2.

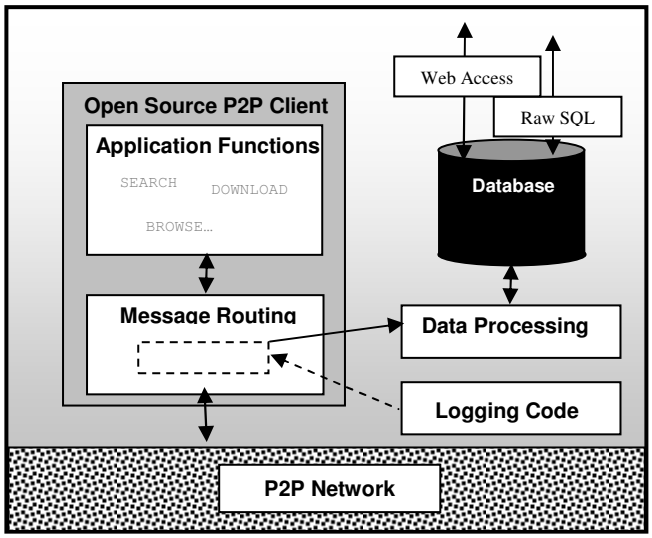

Figure 2. System Architecture

In order to ensure that sufficient data is gathered, the system is capable of maintaining a large number of network connections, for example by connecting as an Ultrapeer when monitoring Gnutella. Furthermore, in order to ensure data is representative, the system periodically re-connects to different areas of the P2P network.

5.2. Maintaining User Anonymity

Open publication of IP addresses and other identifying data is ethically dubious and would likely have a number of undesirable effects. Furthermore, studies have suggested that P2P users are migrating to those file sharing systems which are more difficult to monitor [20]. It is therefore likely that publication of user data from one P2P system would drive users to other, unmonitored systems or perhaps even result in the P2P community attempting to exclude the tracing client. Recent research [20] has also suggested that the level of perceived anonymity offered by P2P networks has a significant effect on user behaviour. This implies that the publication of IP addresses might cause a significant 'observer effect'.

While maintaining anonymity is desirable, a globally unique user identifier (GUID) is often required to accurately track the behaviour of users over time. For this reason, as data is gathered, all IP addresses and user-names are switched for a randomly assigned GUID. Any additional information encapsulated in the original identifier, such as country and service provider, is resolved and stored separately in the database.

Replacing real world identifiers with a randomly assigned but consistent GUID prevents third parties from associating trace data with individuals. However, in the long term this method would lead to the accumulation of data on millions of P2P users, which gives rise to significant security implications. We have therefore arrived at a compromise solution, wherein we only attempt to ensure that GUIDs remain unique during a typical period of connection (session), after which time the IP/GUID mapping is discarded and, if that peer is observed again, it will be assigned a new GUID.

This compromise between maintaining anonymity and user tracking is evaluated in section 6.

5.3. Data Collection and Storage

Due to the scalability problems associated with resource discovery on decentralized P2P networks, P2P systems have increasingly moved towards Super-node architectures such as the architecture used in Kazaa [13] or the Gnutella 0.6 ultra-peer scheme [12]. Concurrently, the scalability problems which arise from the use of a single indexing server have prompted centralised systems to move towards more decentralized architectures that utilize user-hosted indexing servers as demonstrated by DirectConnect and eDonkey. In both cases, the presence of peers on the application-level network which are responsible for routing a greater proportion of messages facilitates application-level monitoring. By connecting to the network as a Gnutella 'ultra-peer', a Direct Connect 'hub' or eDonkey 'server', a greater proportion of traffic can be captured using passive application-level monitoring.

The Bittorrent network is a special case. As Bittorrent does not support resource discovery, torrents are indexed on publicly available trackers, which are accessible on the web. Thus tracing of Bittorrent may be achieved by querying the trackers for details of users currently participating in each per-file

overlay. An overview of this tracing methodology is described in more detail in [36].

As we intend that tracing data should be made accessible to a broad audience, we will use a standard MySQL database for data storage. As SQL is currently the most popular database technology for online applications we hope this will maximize the accessibility of the system. A separate SQL database is maintained for each P2P system being monitored and each of these databases contains per-message tables. Each message that is stored in the database is time-stamped, facilitating the retrieval of data for a specific instant or time-period. In order to maintain flexibility, the system also logs all message types as it is difficult to predict in advance what data may be of interest to other researchers

5.4. Data Access and Presentation

Alongside raw SQL access, we also provide a web-based method of data access for interested parties. We hope this will allow the system to support a range of users with diverse requirements. We envision that three classes of user will make use of the system: i) corporate users, ii) computing researchers and iii) non-computing researchers.

Corporate users of the system might include P2P developers, who could use the system to assess the market penetration of their P2P products, and the music and film industry that might use the system to assess the extent to which their products were being distributed on P2P systems. To facilitate access for corporate users in particular, the system supports on-the-fly generation of common graphs illustrating both current and historic data based on a number of criteria including: P2P client popularity, file popularity and availability, level of user participation and free-riding. The system is also capable of exporting this same data in common formats such as comma separated value (CSV) files and Excel (XLS) spreadsheet documents. To further facilitate the association of P2P traffic with real-world factors, graphical data is annotated with news articles containing references to P2P, which are culled from RSS feeds. This functionality may be used to answer questions such as whether high-profile copyright prosecutions increase levels of free-riding, or whether news about a specific P2P client affected its level of use.

Computing researchers are most likely to be interested in accessing raw traffic data provided by the system. This is possible through direct access to the SQL database which allows more versatility in interrogation than hard-coded trend data that the system provides.

Non-computing researchers are supported by the systems ability to export traffic data in CSV and XLS formats, which can both be accessed using common office software. It is also possible that 'casual' Internet users may find this data of interest, though the requirements of these users have not been explicitly considered in the design of the system. The web interface is shown in Figure 3.

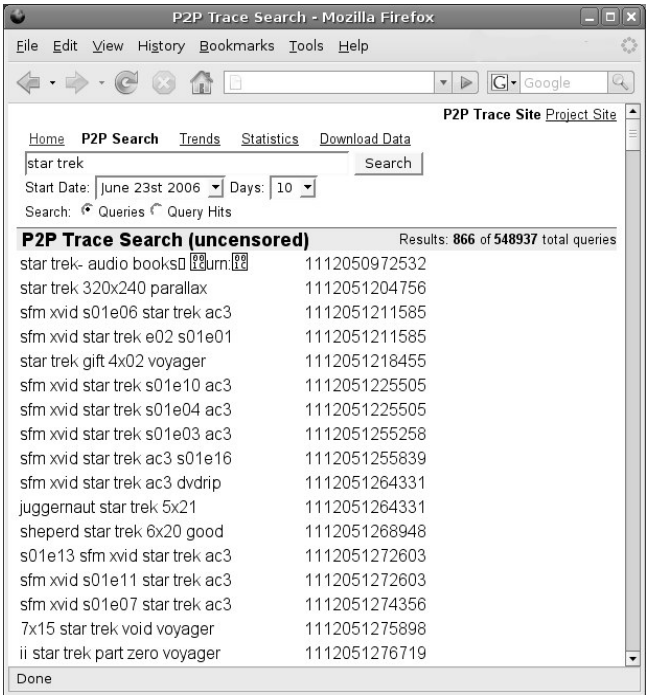

Figure 3. Web Interface of the Open Tracing System

5.5. Implementation Status and Access

The current implementation of the Open Tracing System focuses on the tracing of the Gnutella network, the results of which are being used as a basis to evaluate system functionality (as will be discussed in section 6). Adding support for tracing additional networks is being implemented in parallel to this.

The system is currently at a pre-alpha stage and therefore access to it must currently be arranged through the authors of this paper. However, we are actively looking for case studies, such as those described in section 7, which we hope will guide system development. We anticipate that, in due course, the open P2P tracing system will be made freely accessible online.

6. Initial evaluation results

We have begun analyzing the performance of the Open P2P Tracing System in terms of its network, computational and storage requirements. The system is hosted and evaluated on a 2.8GHz Intel P4 with 512MB RAM and a 100GB hard drive connected to the Internet via a high-speed academic network.

In order to minimize invasiveness during evaluation, the modified tracing peer maintains a single ultra-peer connection and allows unlimited incoming leaf-node connections. As previously described, in order to ensure the typicality of our trace, the system periodically reconnects to the network at an interval of six hours. This figure was derived empirically – we found that a shorter time resulted in connections to less stable peers and less volumes of data, whilst a longer time introduced local effects (e.g. the sharing preferences of core peers) that could impact on the data.

6.1. Networking Requirements

The local network requirements of tracing Gnutella have been assessed through experimentation, while gathering trace data. This reveals that the system consumes an average bandwidth of 98kbps as a result of routing resource discovery messages and an additional 9kbps due to routing control messages, which is commensurate with results obtained elsewhere [7]. The networking requirements of passive application level tracing can easily be met by our available networking infrastructure.

6.2. Storage Requirements

The storage requirements of our tracing methodology were assessed during the gathering of a single-connection Ultrapeer trace of the Gnutella network, conducted over a period of one month. Experimental results are shown in Figure 4.

The storage requirements of tracing the Gnutella network using MySQL's standard data compression range from a minimum of 40MB per day to a maximum of 95MB per day. While this makes long-term tracing feasible using standard desktop storage hardware, available storage capacity still forms the bottleneck in our tracing capability and for this reason, only one tracing connection per monitored network will be maintained by the Open Tracing System for the immediate future.

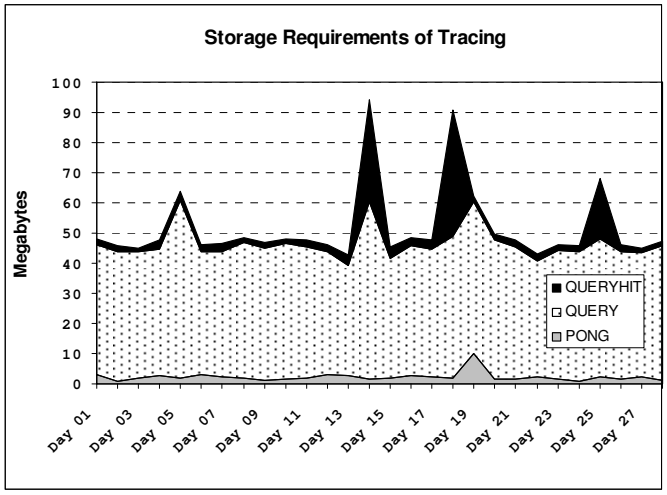

Figure 4. Storage Requirements of Tracing

6.3. Anonymization

As previously discussed, the anonymization approach used is a compromise between storing large volumes of user records and providing a consistent GUID to support session tracking. During our month long trace of the Gnutella network, we performed a number of experiments to determine an optimal IP discard time.

We first monitored session lengths across our trace and found that more than half lasted less than one hour and that more than 80% less than two hours, this is commensurate with results obtained elsewhere [10]. Figure 5 shows the relationship between IP discard time and the percentage of sessions where any data would have been lost. The 'long tail' of the graph shown in Figure 5 is due to the presence of a small number of highly available peers with server-like characteristics and implies that total session coverage would require an unfeasibly long ID-discard period, in turn leading to the maintenance of very large numbers of IP addresses.

Figure 6 explores the relationship between discard time and the number of IP addresses stored by the system. The graph shows that the number of stored IP's varies significantly over the period of our trace and based upon the discard time used. Based upon these results, we have selected a discard time of 6 hours. This period successfully captures 93% of sessions as shown in Figure 6 and results in the open tracing system storing an average of fewer than 800 IP addresses at any one time as shown in Figure 6.

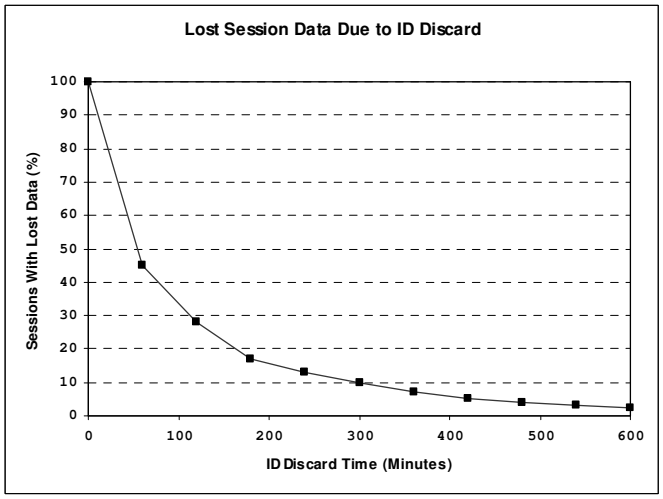

Figure 5. Effect of ID Discard Period on Lost Session Data

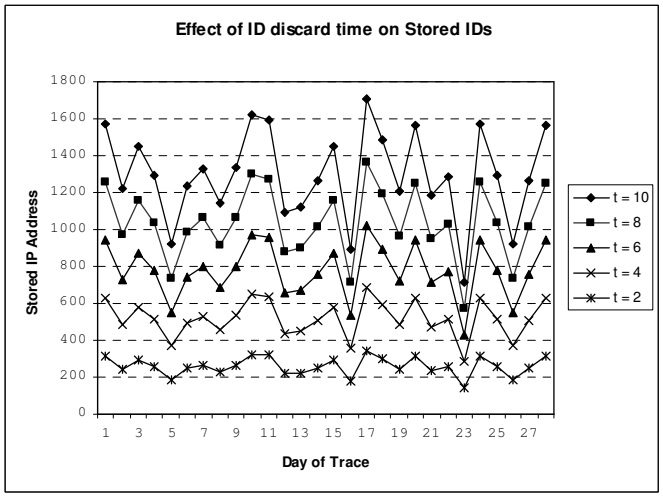

Figure 6. Effect of Discard-Period on Number of Stored IPs

7. Exploiting Open Tracing

In order to better illustrate how an open P2P tracing system may be used to expand our understanding of how P2P file sharing systems are used, this section provides a detailed summary of studies that have been performed by Lancaster University using the same passive application-level tracing methodology as used in the open tracing system.

"Free Riding on Gnutella Revisited: the Bell Tolls?" revisited Adar's 2001 study of Gnutella traffic. Adar found that (i) over two thirds of Gnutella users download files while not uploading – effectively free riding and (ii) user contribution was highly asymmetric with the top contributors providing a disproportionate share of files. Our 2005 study found that in the intervening four years, this situation had worsened. While we found similarly asymmetric contribution levels, the number of contributing peers had fallen by more than half. The stark difference between these data points shows the benefit of revisiting established data points, a key goal of the open tracing system.

"Is Deviant Behaviour the Norm on P2P File Sharing Networks?" was performed in 2005 in collaboration with psychologists, to answer the question of whether the perception of anonymity and lack of censorship on Gnutella encourages the distribution of illegal pornographic material. Our study found that while the level of resource discovery traffic relating to illegal pornographic material was high (1.6% of searches and 2.4% of responses), that this traffic was generated by a small, yet active subset of the network that shared nothing else. This finding has useful implications for the policing of P2P file sharing networks, but more importantly shows how collaboration between academic disciplines, another key goal of the open tracing system, can lead to better understanding of P2P user behaviour.

"Supporting Law Enforcement in Digital Communities through Natural Language Analysis" revisited our 2005 trace data, looking specifically at the level of traffic related to child abuse material. We found that 1% of search traffic related to such material and 1.6% of response traffic. Building upon this analysis in collaboration with linguistics researchers and child protection researchers and professionals, this paper suggested an approach to automating the policing of P2P file sharing systems for child abuse material. As with the previous study this illustrates the significant benefits of making P2P file sharing trace data to other disciplines.

8. Future Work and Conlusions

This paper has highlighted significant shortcomings in the existing body of work on P2P monitoring, and described the implementation of a large-scale, open and ongoing trace that can be freely accessed by researchers from diverse backgrounds. Based upon an extensive review of existing P2P studies, we have selected a non-invasive tracing methodology that we will incrementally apply to five of today's most popular P2P file sharing networks. At the current time, tracing functionality has been implemented for the Gnutella network and evaluation of the system shows that our methodology is capable of gathering, anonymizing and logging Gnutella traffic in real-time using standard desktop hardware. The system facilitates access for users from diverse backgrounds- a direct interface to the SQL database allows versatile access for computing researchers, while a simplified web interface and on-the-fly computation of common P2P characteristics such as the level of 'free riding' and relative file-type popularity facilitate access for those from non-computing fields.

In the short term, future work will focus on the implementation of tracing functionality for additional P2P systems. In the longer term we intend to investigate incorporating Natural Language Processing mechanisms into the system to allow the user to perform more sophisticated analyses. In addition to this we will also examine the feasibility of using technologies such as Aspect Oriented Programming to assist in the non-invasive monitoring of P2P systems, and also to investigate alternative, more scalable data storage solutions.

In parallel to extending tracing support, we intend to evaluate the usefulness of the system as a tool, using a number of case studies. Part of this will include working with psychology researchers to investigate the process of group formation in P2P communities. This will build upon our previous work [5] [8] [28] [33] and allow us to explore the extent to which the system can support inter-disciplinary research.

9. References

[1] Merriden T., Irresistible Forces: the Business Legacy of Napster and the Growth of the Underground Internet, Capstone Publishing, 2001, pages 1-11.

[2] Saroiu S., Gummadi K., Dunn R. J., Gribble S. D., Levy H. M., An Analysis of Internet Content Delivery Systems, in the proceedings of the 5th International Symposium on Operating Systems Design and Implementation (OSDI), San Francisco, CA, 2004, pages 315-327.

[3] Gummai K., Dunn R. J., Saroiu S., Gribble S. D., Levy H. M., Zahorjan J., Measurement, Modeling and Analysis of a P2P File-Sharing Workload, in the proceedings of the 19th Symposium on Operating Systems Principles (SOSP'03), Bolton Landing, NY, 2003, pages 314-329.

[4] Hughes D., Coulson G., Walkerdine J., Free Riding on Gnutella Revisited: the Bell Tolls?, in IEEE Distributed Systems Online, volume 6, number 6, 2005. sdl2.computer.org/comp/mags/ds/2005/06/o6001.pdf

[5] Hughes D, Walkerdine J., Lee K., Monitoring Challenges and Approaches for P2P File Sharing Systems, in the proceedings of the 1st International Conference on Internet Surveillance and Protection (ICISP'06), Cap Esterel, France, 2006, page 18-18.

[6] Plonka D., Napster Traffic Measurement, University of Wisconsin-Madison, 2000, available online at: http://net.doit.wisc.edu/data/Napster

[7] Adar E., Huberman B., Free Riding on Gnutella, First Monday, volume 5, number 10, October 2000, available online at: www.firstmonday.org/issues/issue5_10/adar/

[8] Hughes D., Gibson S., Walkerdine J., Coulson G., Is Deviant Behaviour the Norm on P2P File Sharing Networks?, in IEEE Distributed Systems Online, volume

7, number 2, 2006. csdl.computer.org/comp/mags/ds/2006/02/o2001.pdf

[9] Ripeanu M., Iamnitchi A., Foster I., Mapping the Gnutella network, published in IEEE Internet Computing., volume 6, number 1, 2002, pages 50-57.

[10] Saroiu S., Gummadi K., Gribble S. D., Measuring and Analyzing the Characteristics of Napster and Gnutella Hosts, published in Springer-Verlag Multimedia Systems volume 9, number 2, 2003, pages 170-184.

[11] Chu J., Labonte K., Levine N., Availability and locality measurements of peer-to-peer file systems, in the proceedings of ITCom: Scalability and Traffic Control in IP Networks, Proceedings of SPIE, volume 4868, Boston, MA, 2002, pages 310-321.

[12] The Gnutella Protocol Specification v 0.6, available online at: rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html

[13] Kazaa homepage, available online at: www.kazaa.com

[14] eDonkey homepage, available online at: www.edonkey2000.com

[15] Direct Connect homepage, available online at dcplusplus.sourceforge.net

[16] Cohen B., Incentives Build Robustness in Bittorrent, available online at: http://www.bittorrent.com/bittorrentecon.pdf, May, 2003.

[17] Jtella homepage, available online at: jtella.sourceforge.net

[18] Open DirectConnect homepage, available online at: sourceforge.net/projects/odc/

[19] Azureus home page, available online at: azureus.sourceforge.net/

[20] Karagiannis, T., Broido, A., Brownlee, N., Faloutsos, M., Is P2P Dying or Just Hiding?, In the Proceedings of Globecom'04, Dallas, TX, 2004, pages 1532-1538.

[21] Qiao Y., Bustamante F.E., Structured and Unstructured Overlays Under the Microscope - A Measurement-based View of Two P2P Systems That People Use, published in the Proceeding of the USENIX Annual Technical Conference, Boston, MA, 2006, pages 10-10.

[22] The Gnutella Protocol Specification v0.4 (Document Revision 1.2), available online at: http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf, 2001.

[23] Subhabrata S., Spatscheck O., Wang D., Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures, in the proceedings of the thirteenth international world wide web conference (WWW2004), New York, NY, 2004, pages 512-521.

[24] AT&T Gigascope homepage, available online at: http://public.research.att.com/viewProject.cfm?prjID=129

[25] The Onion Router (Tor) homepage, available online at: http://www.torproject.org/index.html.en

[26] Bauer K., McCoy D., Grunwald D., Kohno T., Sicker D., Low-resource Routing Attacks Against Tor, in the proceedings of the 2007 ACM workshop on Privacy in Electronic Society (WPES'07), Alexandria, VA, 2007, pages 11–20.

[27] McCoy D., Bauer K., Grunwald D., Kohno T., Sicker D., Shining Light in Dark Places: Understanding the Tor Network, in the proceedings of the 8th Privacy Enhancing Technologies Symposium (PETS'08), Leuven, Belgium, 2008, pages 63-76.

[28] Walkerdine J., Hughes D., Lee K., The Effect of Viral Media on Business Usage of P2P, in the proceedings of the 7th international IEEE conference on Peer-to-Peer Systems (P2P'07), Galway, Ireland, 2007, pages 249-250.

[29] Ritter J., Why Gnutella Can't Scale, No Really, available online at: http://www.darkridge.com/~jpr5/doc/gnutella.html.

[30] SProbe homepage, available online at: http://sprobe.cs.washington.edu

[31] Izal M., Urvoy-keller G., Biersack E., Felber P., Al Hamra A., Dissecting BitTorrent: Five months in a torrent's lifetime in the proceedings of the Passive and Active Measurement workshop (PAM'04), Antibes Juan-les-Pins, France, 2004, pages 1-11.

[32] Thommes R., Coates M., BitTorrent Fairness: Analysis and Improvements, in the proceedings of the 4th Workshop on the Internet, Telecommunications and Signal Processing (WITSP'05), Noosa Heads, Australia, 2005, available online at: http://www.tsp.ece.mcgill.ca/Networks/projects/pdf/thommes_WITSP05.pdf

[33] Hughes D., Rayson P., Walkerdine J., Lee K., Greenwood P., Rashid A., May-Chahal C., Brennan M., Supporting Law Enforcement in Digital Communities through Natural Language Analysis, in the proceedings of the 2nd International Workshop on Computational Forensics (IWCF'08). Washington D.C., USA, 2008, pages 122-134.

[34] Piatek M., Kohno T., Krishnamurthy A., Challenges and Directions for Monitoring P2P File Sharing Networks – or– Why My Printer Received a DMCA Takedown Notice, available online at: http://dmca.cs.washington.edu/dmca_hotsec08.pdf

[35] Loesing K., Sandmann W., Wilms C., Wirtz G., Performance Measurements and Statistics of Tor Hidden Services, in the Proceedings of the International Symposium on Applications and the Internet (SAINT), Turku, Finland, July 2008, pages 1-7.

[36] Peer Guardian home page, available online at: http://phoenixlabs.org/pg2/

[37] Hughes D., Lee K., Walkerdine J., An Open Tracing System For P2P File Sharing Systems, published in the proceedings of the second International Workshop on P2P Systems and Applications (P2PSA '07), Morne, Mauritius, May 2007, pages 3-9.