Analysis of BitTorrent Peers’ Behavior and Monitoring Trends
(Category: Law and security)

Camilo Andrés González Toro
M.Sc. in Internet Software Systems
(DOB: May 18th, 1985)
Email: camigonz@gmail.com

Supervised by:
Dr. Tom Chothia
Lecturer
Email: T.P.Chothia@cs.bham.ac.uk

School of Computer Science
University of Birmingham
Birmingham, United Kingdom
September, 2009

Copyright © 2009, School of Computer Science, University of Birmingham
Table of Contents

Abstract ...............................................................................................................................1
1. Introduction.............................................................................................................................2
2. Background on BitTorrent .................................................................................................3
  2.1 Definition ..........................................................................................................................3
  2.2 BitTorrent Entities ............................................................................................................3
  2.2.1 Initial Seeder ................................................................................................................3
  2.2.2 Metainfo file .................................................................................................................3
  2.2.3 Torrent Servers ...........................................................................................................4
  2.2.4 BitTorrent Trackers ....................................................................................................4
  2.2.5 Peers ............................................................................................................................4
  2.3 The BitTorrent Protocol ...............................................................................................4
  2.3.1 Initial File Serving .......................................................................................................4
  2.3.2 Peer Arrival ................................................................................................................5
  2.3.3 File Distribution Process ...........................................................................................6
3. Previous Work ....................................................................................................................8
4. Methodology .....................................................................................................................10
  4.1 Development of the Instrumented Client – BitTorrent Explorer ...................................10
  4.1.1 BitTorrent Explorer Components .............................................................................11
  4.1.2 BitTorrent Explorer Processing Architecture ..........................................................12
  4.1.3 Anonymity using Tor ...............................................................................................14
  4.2 Client Testing, Tuning and Feedback ............................................................................15
  4.3 Data Gathering ...............................................................................................................15
  4.4 Data Analysis – From data to Peer Properties and Behavior .........................................16
5. Peer Behavior ....................................................................................................................17
  5.1 Common Peer Behavior ...............................................................................................19
    5.1.1 Popularity ................................................................................................................19
    5.1.2 Ports reported .........................................................................................................20
    5.1.3 Peer Connection Time ............................................................................................21
  5.2 Summary of Common Peer Behavior ...........................................................................22
6. Deviant Peers and Monitors .............................................................................................23
  6.1 Deviant Behavior ..........................................................................................................24
  6.2 Detection of Monitors, Justifiable and Non-Justifiable Deviant Behavior .......................24
  6.3 Other Possible Monitoring Trends ................................................................................26
7. Conclusions ......................................................................................................................28
8. Important Notes ................................................................................................................29
9. References .........................................................................................................................30
Abstract

Methods to classify peers and detect deviant peer behavior have started to emerge for the BitTorrent network using sets of heuristics, based on analysis of Autonomous Systems and IP-range behavior, and measurements made from network traces. The present project presents a set of heuristics that is based on Peer Behavior which allows the classification of peers, dividing them on Common and Deviant Peers. In addition, the possibility of detecting monitoring agents is presented based on the information resulting of the peer classification. Finally, some possible monitoring trends that may be used or may be being used by monitoring agents are mentioned.
1. Introduction

With the appearance of new Peer-to-Peer (P2P) networks and protocols, users around the world are now able to share different kinds of content without relying on a central server and in a way such that every peer provides/shares its own resources (e.g. data, processing power, memory, etc.).

In this sense, peers are able to share information without any central point of governance, introducing problems to law enforcement agencies, like an increase in content piracy and copyright infringement, and an increased difficulty to monitor peer’s activity.

Hence, monitoring organizations (like copyright holders, law enforcement agencies and others) are looking for ways to enforce the law or even only to perform studies about the BitTorrent network; however, they face different challenges as documented by Hughes, et al., (1). To monitor the BitTorrent network and to start surpassing these challenges, monitoring users may use different approaches which may expose these monitoring clients, establishing the presence of both normal BitTorrent users and other kinds of users (e.g. monitoring clients) in the BitTorrent network as discussed by Siganos, et. al. (2).

In addition, recent news about torrent sites in discussions and with involved in legal actions with the music and film industry and other news (3) that mention warning messages from the MPAA which read “Stealing movies leaves a trail”, it is clear that monitoring must be being performed by different organizations. This opens questions like establishing who the monitors are and whether or not it is possible to detect the monitoring clients.

The present study shows an analysis of peer behavior that can be used to classify peers, allowing automatic classification of deviant and regular peers in BitTorrent, one of the most currently used P2P protocols. The study is based on empirical evidence gathered from which common peer observed behavior can be obtained and leads to the classification of peers, and is meant to be used, in conjunction with related work, to further extend the peer classification categories with the use of different kinds of empirical measures.

A background on the BitTorrent protocol is presented in Section 2, followed by a study on previous work in Section 3. Section 4 presents the methodology of the present study, ranging from the conception and development of the data gathering tool to information about the data gathered. Section 5 details information from the data analysis performed, showing a profile of common peer behavior. Section 6 covers the data analysis of deviant behavior and monitoring trends, finally concluding with Section 7.
2. Background on BitTorrent

2.1 Definition

BitTorrent was proposed by Cohen (4) as a file distribution system which works by making use of a peer-to-peer protocol (namely the BitTorrent Protocol) and tries to achieve Pareto efficiency\(^1\) using a tit-for-tat method where peers upload content to peers from where content is also being received.

2.2 BitTorrent Entities

To provide further details about BitTorrent, it is useful to establish the entities which form it. Figure 1 illustrates three of the five main entities, and the communication protocol that is used across each of them. The following subsections briefly detail each of the entities found in the BitTorrent protocol.

![Active BitTorrent Entities and Communication Protocols](image)

Figure 1. Active BitTorrent Entities and Communication Protocols

2.2.1 Initial Seeder

The initial seeder is the BitTorrent Peer that initiates the file sharing process. In this sense, this is an Active Entity\(^2\) which in addition to performing the same activities of a regular peer (see Section 2.2.5) it will have the responsibility of generating the metainfo file (see Section 2.2.2), performing the initial distribution/publication of the metainfo file and sharing the file as the initial seeder.

2.2.2 Metainfo file

Sometimes known as the torrent file due to its .torrent file extension, the metainfo file can be seen as a Static Entity of BitTorrent which contains data about the file or

---

\(^1\) In Computer Science, it can be defined as a “local optimization algorithm in which pairs of counterparties see if they can improve their lot together” which can tend to reach global optima (4).

\(^2\) It is possible to classify entities as Active and Static (or Passive), where Active entities are those that perform any kind of processing in contrast to Static entities which don’t perform any actions at all, e.g. metainfo files.
files that are being shared. It includes data that is used to locate tracking servers, to perform data integrity checks, and from the data contained in the metainfo file a SHA1 hash which uniquely\(^3\) represents the file being shared.

### 2.2.3 Torrent Servers

As defined in the BitTorrent technical framework, the metainfo file should be put on an ordinary web server (4) such that other peers may be able to acquire it. In this sense, the Torrent Servers are ordinary web servers that provide a place from where the BitTorrent users can download the metainfo files.

### 2.2.4 BitTorrent Trackers

Although all peers share the same metainfo file, they don’t have any way to know which other peers are involved in the distribution of a specific file. As defined by Cohen (4) “trackers are responsible for helping downloaders find each other”, i.e., tracking servers are active BitTorrent entities that keep information about which peers are currently sharing a given file. All the peers that are registered in the tracker and which are downloading the same file are known to belong to the same swarm of peers.

### 2.2.5 Peers

Peers, which can be seen as BitTorrent clients sharing/distributing the same file (5), are active BitTorrent entities which can be classified as a) leechers which are peers that haven’t got the full copy of the file being distributed and b) seeders which are peers that have downloaded the file in its whole.

Peers are in charge of contacting the tracker to acquire a set of peers sharing the same file, to communicate with other peers in order to distribute the file across them, and to share control information to optimize their own download/upload rates using the tit-for-tat mechanism described in (4) (6) (5).

### 2.3 The BitTorrent Protocol

In the previous section the different BitTorrent entities were defined. In this section the main interactions between these entities will be described.

### 2.3.1 Initial File Serving

---

\(^3\) Subject to the mathematical constraints of the SHA1 secure hash functions.
In order to initiate a file distribution process using BitTorrent, the Initial Seeder must generate a metainfo file that contains data about the file that will be distributed (see Figure 2). Specifically, values like the announce field, announce-list field and the infohash which is calculated based on the data in the metainfo file are of great importance for setting up the P2P network.

![Diagram](image)

**Figure 2. Metainfo File Generation**

After the metainfo file is generated, it must be published so that other users can join the swarm of peers that are distributing the file, and the initial seeder must join the swarm by reporting to the tracker(s) informing that it has a complete version of the file\(^4\).

2.3.2 **Peer Arrival**

After the metainfo file has been made available, other users that would like to get a copy of the file being distributed, and hence, join the swarm of peers, can download a copy of the metainfo file to get the information needed to join the swarm, i.e. the infohash, the announce and announce-list fields and the total file length.

Once the infohash is calculated, the new arriving peer can contact the tracker(s) to get a set of the peers which are currently in the swarm sharing the same file. As shown in Figure 1, this communication is made using the BitTorrent Protocol specification and runs over HTTP/HTTPS, referred as the Tracker HTTP/HTTPS Protocol (5). A sample HTTP announce request url has the following format:

```plaintext
http://tracker.host.com/announce?info_hash=[encoded_20_byte_infohash]
&peer_id=[encoded_20_byte_peer_id]&port=[port_number]&uploaded=x&downloaded=y&left=z
```

\(^4\) This behavior can be modified if desired (e.g. super-seeding).
When the set of peers is acquired from the tracker(s), the new arriving peer can connect to other peers using the Peer Wire Protocol (5). As an example, Figure 3 shows the case of the first arriving leecher downloading the metainfo file, retrieving a set of peers from the tracker(s) which in this case only contains the ip and port of the initial seeder, and finally connecting to the initial seeder which has the full copy of the file (pieces 1, 2 and 3).

2.3.3 File Distribution Process

After two or more peers have joined a swarm to distribute a specific file, communication is directly held between peers without the need of a central server, using the Peer Wire Protocol, which begins with a handshake message where peers can verify that they are both interested in the same file by transmitting the infohash, and if the handshake is correct, peers will exchange length-prefixed messages to perform the communication that enables the distribution of the file across the P2P network.
Figure 4 shows an example of a BitTorrent peer swarm with 5 peers (2 seeders and 3 leechers) and the interaction of file pieces between the peers. To achieve this, leechers send a request message to other peer(s) asking for some piece of the file. Pieces may be requested to other leechers that have this piece of the file or to seeders. It is important to note, that peers are in charge of attempting to optimize their own download rate, which is achieved using a choking algorithm as described in Cohen (4).

\[5\] For simplicity; the exact behavior states that a peer requests a block of data from a piece of the file.
3. Previous Work

Different studies about the BitTorrent Protocol have been done over the lapse of the last years, ranging from mathematical modeling to empirical studies. In terms of mathematical modeling, Guo, et al. (7), introduce a modelling, including mathematical equations that can be used to estimate the lifespan of a given torrent, number of successful peers and number of peers that fail to download the file due to lack of seeds.

Pouwelse, et.al. (8) perform an analysis over an eight months trace, presenting detailed analysis about four performance measurements of the BitTorrent network, including availability, integrity, flashcrowd effect and download performance of the system. It also brings important details to take into account for the development of BitTorrent clients, like highlights on some availability issues.

In 2006, Hughes, et al. (1), illustrate a set of P2P network traces from diverse studies, explaining and describing the different monitoring approaches that can be followed, and associating each study with the corresponding monitoring approach used. Additionally, a set of monitoring challenges is presented and some highlights to the importance of developing high quality empirical studies of P2P systems is done by the authors.

Piatek, et. al. (9) introduce additional challenges and provide directions for monitoring P2P systems. In their study, they present two monitoring options (indirect vs direct\(^6\)) and provide empirical evidence showing that some law enforcement agencies are using indirect monitoring. By making use of the misreporting client attack which “allows malicious clients to frame arbitrary IPs for infringement via a simple HTTP request” they successfully framed innocent IP’s (assigned to desktop computers, printers and wireless access points) which received copyright (DMCA) takedown notices, showing that indirect monitoring is less conclusive than direct monitoring.

In 2009, Siganos, et. al. (2) begin their work based on the fact that the BitTorrent networks is populated by different kinds of clients and provide initial heuristics oriented towards the automatic detection of those clients that do not behave as standard clients, i.e., deviant clients. The heuristics presented are based on analysis at the level of Autonomous Systems (AS) and at the level of IP prefixes.

The present study, following the Siganos, et. al. research, tries to provide additional heuristics to complement those already provided looking for the possibility of an automatic classification of peers based on observed behavior. In contrast to the

---

\(^6\) These will be covered in more detail later in this report.
previous papers, this study focuses on behavior at the peer [IP + port] and IP levels, providing a peer classification based on the metrics defined and showing some of the results obtained using the metrics defined, including information about detected monitors.
4. Methodology

The present study follows a similar methodology to the one used by Pouwelse, et. al. (8), commencing by designing and developing an instrumented client, executing the client to obtain an application-level trace (1) of BitTorrent activity, and performing data processing and analysis. The following sections will cover in more detail the activities mentioned.

4.1 Development of the Instrumented Client – BitTorrent Explorer

Based on the entities mentioned in section 2.2, five tasks were defined to monitor the BitTorrent protocol:

1) Download recently published or featured torrent metafiles from the Torrent Servers (e.g. Mininova or ThePirateBay).
2) Extract the trackers (announce and announce list fields) from the metafiles and record them. Record other important fields like the infohash\(^7\), length of the file, and others.
3) Contact the trackers to retrieve peers that are in a swarm.
4) Perform tracker scrapes using the scrape mechanism.
5) Contact selected peers to perform a handshake and listen to sent messages for an established amount of time.

With these tasks in mind, five software components were established as depicted in Figure 5 below. These components are the base of the monitoring tool, known as BitTorrent Explorer, which was developed and used in this project. The development of some of these components is based on the Snark Project (10), a BitTorrent Client based in Java. The source code was updated to comply with the current BitTorrent Protocol specification.

\(^7\) Infohash value is calculated as the SHA1 hash of the info field in the metainfo file
4.1.1 BitTorrent Explorer Components

a) **Torrent Crawler:** Executes HTTP requests to gather newly published metainfo files that are located in torrent servers like Mininova/PirateBay.

b) **Peer Retriever:** Contacts the trackers gathered by the Torrent Crawler (announce url’s) as established by the BitTorrent Protocol to obtain peers (IP+port pairs) which belong to different swarms. The Peer Retriever component is also in charge of contacting the trackers via the scrape url to obtain the number of seeders and leechers in each swarm.

c) **Peer-2-Peer Client:** Contacts the peers gathered by the Peer Retriever, performs a Handshake and listen for messages for a set amount of time. It is possible to specify the list of peers to contact if required, to avoid contacting all the peers due to high consumption of resources.

d) **IP Data Service:** Loads additional information of the IP’s that have been acquired by the Peer Retriever component. This includes IP-to-Country mapping, blacklists loading and others.

e) **BitTorrent Explorer Coordinator:** In charge of distributing the tasks mentioned above across different computing nodes and scheduling the tasks (e.g. Crawling for Torrents each 15 minutes). The Coordinator is formed up by a configuration

---

8 Also works over a single node
and central Controller, a Coordination Server and Coordination Clients which receive tasks from the Coordination Server that need to be performed.

4.1.2 BitTorrent Explorer Processing Architecture

Once the BitTorrent Explorer is initiated, the Controller will load a given configuration file that allows the setting of different parameters, including which tasks should be performed (i.e. it may be desirable to keep some of the components turned off for given tests), ports and proxies to be used amongst others.

In the general case, i.e. assuming all the components are set to be active in the configuration file, the Controller will create a new Coordination Server which will be in charge of starting up the other components at given time intervals. In this sense, the Coordination Server can be configured to initiate Peer Retrieving operations each 15 minutes, distributing the work load across the different Coordination Clients that are connected to the server.

After the Coordination Server initiates the additional tasks, each of them will be run independently of each other, gathering required data from a database (see Figure 6), and logging the information gathered from the BitTorrent monitoring tasks into the same database. The flow of data from and into the database can be used to explain the processing architecture of the BitTorrent Explorer instrumented client as follows:

Initial Data in the Database:

Tables bt_torrent_server (e.g. ThePirateBay.org) and bt_torrents_page (e.g. top 100 featured movies torrents in ThePirateBay.org, top 100 featured music torrents in ThePirateBay.org) contain data about the torrent sites that are meant to be contacted by the tool during the trace gathering. Ideally, these tables will contain these data before running the instrumented client.

Torrent Crawling:

Initiated by the Coordination Server after the tool is executed, the Torrent Crawler component will query the data stored in the two tables specified above and will make an HTTP request to the specified pages. Each response from the servers will be parsed, looking for links that point to metainfo files. Each of these metainfo files will be downloaded to the server and will be processed to gather data like announce

---

9 This holds for the Torrent Crawler, Peer Retriever and P2P Client components. The IP Data Service is meant to be used as a post processing step. The three former components can be used/tested independently from the Coordination Server if required.

10 E.g. in the case of ThePirateBay, links that comply with the Java RegEx: http://torrents.thepiratebay.org/\(\d+/(.\d+\.torrent\]
and announce-list fields which are used to populate the bt_tracker and tracker_file tables of the database, and data that will be used to populate the bt_metafile table, including the infohash and length fields of the metainfo file.

![Figure 6. Database model for data gathering stage](image)

**Peer Retrieving:**

The Peer Retriever component will contact the BitTorrent trackers using the announce urls stored in the tracker_file table. Upon successful reply from the trackers to the announce requests, information about the peers will be stored in the database, storing the IP in the ip_data table, the port number associated with this IP in the peer table, and creating an association between the peer and the tracker_file in the swarm_peer table.

If an IP or peer (IP+port pair) gets reported more than once, the system will not create duplicates in the database. For the case of a peer, the first and the last time it was reported will be recorded.

Additionally the component will also request the scrape information from the BitTorrent trackers making use of the scrape url mechanism to obtain the number of leechers and seeders in a swarm. These data will be stored in the scrape table.

---

11 Tracker_file stored the associations between the metainfo files and the trackers, such that it is possible to establish which BitTorrent Trackers are tracking a given file.
P2P Client:

If enough computing resources are available, the P2P client can be used to contact and listen to the peers retrieved by the Peer Retriever component. In this sense, the Coordination Server will distribute the work between the P2P clients in each of the computing nodes, assigning to each node a set of data from the swarm_peer table. Each of the P2P clients will contact each of the peers assigned and will log the information about the connection (e.g. messages received, handshake status, etc.).

Because the cost of performing direct P2P connections to thousands of peers is so high (9), the P2P client has also been designed such that a set of peers can be read from a file, allowing direct monitoring of selected peers only, instead of attempting direct monitoring to all of the peers in the database. Information gathered using this mechanism will be recorded in a plain text file that can be processed afterwards.

IP Data Service:

The IP Data Service is meant to be used as a post processing component, which will load additional information regarding the different IP’s gathered during the logging of the trace. Specifically, three different services are offered, including the ability to load IP-to-Country mappings (11), IP blacklists (12), and to establish which of the IP’s are reported as TOR nodes.

4.1.3 Anonymity using Tor

For anonymity purposes the instrumented client was made configurable so that it can route HTTP requests and TCP connections through the Tor\textsuperscript{12} (13) network. Two main points are achieved when using Tor, 1) the node(s) used to monitor BitTorrent using the instrumented client will remain anonymous and 2) as an extension to the first point, enable the possibility of performing the monitoring using a low number of nodes or even using a single node.

Due to the working nature of Tor, where connections are sent encrypted between Tor nodes, the route that the packets take within the Tor network is random, and hence a) the final node that sends the packet to its final destination is random as well and b) the destination node receives the packet from a Tor node and not from the instrumented client nodes, both of the points mentioned above can be achieved.

It is important to note that performance issues have to be considered. The observed performance when the instrumented client is configured to use Tor is much lower than when it is not using Tor. Simple HTTP requests can take up to one minute when

\textsuperscript{12} Specific details about Tor are out of the scope of this document
competing with multiple concurrent requests in Tor, compared to milliseconds when using direct connections. These performance issues were expected and are related to the Tor connection setting times, data encryption and others that must be performed when using Tor.

4.2 Client Testing, Tuning and Feedback

Due to the timeframe of the project, the development and testing tasks were performed highly in parallel, beginning with the update of the Snark BitTorrent Client (10), and continuing with the components mentioned in section 4.1.1. Each of the components and the updated Snark Client were tested independently, ensuring that each individual component was working correctly (unit testing phase).

Once the results from the unit testing phase were satisfactory, including the revision and crosschecking of data against the data acquired by other BitTorrent Clients, the development of the BitTorrent Explorer Coordinator component was started, which is the integration component of the instrumented client. Unit testing was performed in this component to ensure that configuration properties were being read correctly, followed by integration testing in order to review that the system would work correctly as a whole.

During the execution of the integration tests, feedback was gathered regarding different configuration parameters that could be tuned up in order for the tool to perform better. Parameters like connection timeouts, socket reading timeouts, number of concurrent threads and time intervals between successive executions of tasks were tuned up in this stage.

4.3 Data Gathering

Two traces were collected for the purpose of the present study, one of them had duration of one week, beginning on July 21, 2009 at 15:30 and finishing on July 28, 2009 at 16:38 (Trace A), and the other had duration of two days beginning on August 4, 2009 at 21:51 and finishing on August 6, 2009 at 21:48 (Trace B). The second trace was gathered in order to verify information found in the first trace (see Section 5).

Both traces were acquired by monitoring the Top 500 list of ThePirateBay (14), and include the data summarized in Table 1 below. This set of torrents to monitor was

---

13 Although different small traces were collected during the testing period, these two are considered the official traces for the project.

14 Although the first trace has duration of one week, the second trace provides more data in terms of quantity, fact that seems to be related to the performance of the Tor network during the times the traces were gathered.
chosen due to its popularity in one of the most popular torrent sites, hence, providing a greatest set of peers in the swarms monitored in comparison to less popular torrents.

<table>
<thead>
<tr>
<th>Description</th>
<th>Trace A</th>
<th>Trace B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of IP’s</td>
<td>831,039</td>
<td>1,351,853</td>
</tr>
<tr>
<td>Number of Peer’s (IP, Port)</td>
<td>894,529</td>
<td>1,498,015</td>
</tr>
<tr>
<td>Number of Swarm Peers (IP, Port, Tracker)</td>
<td>1,237,380</td>
<td>3,033,892</td>
</tr>
<tr>
<td>Number of Torrents Monitored</td>
<td>967</td>
<td>690</td>
</tr>
<tr>
<td>Number of Trackers Reported</td>
<td>196</td>
<td>181</td>
</tr>
<tr>
<td>Number of Peers Directly Contacted</td>
<td>28,872</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1. Summary of BitTorrent Traces Gathered

4.4 Data Analysis – From data to Peer Properties and Behavior

After the first trace was gathered the data analysis stage was started. To perform the data analysis different peer properties were defined and based on the trace gathered these properties were calculated or estimated depending on the property being analyzed. Section 5 will cover in more detail the Properties that were calculated or estimated from the traces gathered.

From this point, peer behavior can be established based on the measured properties, which allows the setting of observed differences between normal and deviant peer behavior in BitTorrent. In Section 6 an additional analysis of the peers with deviant behavior is performed and a set of heuristics for identifying deviant peers and possible monitoring peers, as proposed by Siganos, et. al. (2), is presented.

---

15 The term Swarm Peer is used to refer to a Peer inside a Swarm. One peer can be associated with multiple swarm peers (i.e. the same peer is reported by different trackers)

16 This represents the numbers of peers that where directly monitored. See Section 5 for further details about Direct vs Indirect Monitoring
5. Peer Behavior

From the data gathered by the instrumented client, some Peer Properties can be identified. Figure 7 presents different Identifiable Properties that can be obtained. Some of these properties can be acquired via Indirect Monitoring and others can be acquired via Direct Monitoring, monitoring options defined by Piatek, et. al. (9).

![Peer Properties Diagram]

Figure 7. Peer Properties

The set of properties that lie under Indirect Monitoring are those that can be established without the need to establish a direct connection with the peer. In this sense, the measurement of these properties is based on the data retrieved from the trackers making use of the announce and scrape url’s.

In contrast, properties that lie under Direct Monitoring require direct connection to the peers. Direct Monitoring provides a richer set of properties, however, compared with Indirect Monitoring the amount of resources required is higher (9), and the difficulty to establish these properties is higher not only in terms of resources but also due to different technical features, like the use of firewalls and NAT’s, the use of non-standard ports, and others reported in previous studies (1) (9).

As shown in Section 4.3, the present study is highly focused on indirect monitoring. A small scale direct monitoring study is performed attempting BitTorrent connections and message listening from 28,872 different peers. In case a BitTorrent connection is successful, the instrumented client will log the bitfield message, if provided, the peer ID and all messages that are received during a lapse of three minutes, after which the connection is dropped in order to free resources to attempt connections to other peers.
Table 2 below provides a description of each of the Peer Properties shown in Figure 7, including information about whether or not the given property was gathered or can be established using the traces gathered by the instrumented client, and the relative difficulty to acquire each property in terms of the technical issues that must be surpassed and in terms of the resources that are required to acquire the given property for a large scale monitoring.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Estimated</th>
<th>Difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peer Popularity</td>
<td>Number of swarms in which the same peer(^{18}) appears</td>
<td>Yes</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>Number of swarms in which the same IP appears</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Number of swarms per IP-Range or AS(^{19})</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Number of files(^{20}) a same peer is downloading</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Number of files a same IP is downloading</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Number of files per IP-Range or AS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peer Popularity by Type</td>
<td>Same as Peer Popularity but limited to a specific set of torrents (e.g. Music, Video, etc.)</td>
<td>No</td>
<td>Low</td>
</tr>
<tr>
<td>Peer Total Connection Time</td>
<td>Total amount of time since the peer was first reported in a swarm or group of swarms until the last time the peer was reported. This property can also be estimated by Direct Monitoring.</td>
<td>Yes</td>
<td>Low</td>
</tr>
<tr>
<td>Peer Blacklisted</td>
<td>Does the peer appears in P2P blacklists such as I-Blocklist</td>
<td>Yes</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>(<a href="http://iblocklist.com/list.php?list=bt_level1">http://iblocklist.com/list.php?list=bt_level1</a>)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peer location and IP properties</td>
<td>Geographic location of the IP (i.e. IP to country map (11)), whois information, whether the IP is reported as a Tor node or not, and others.</td>
<td>Country mapping and Tor identification is done for all peers. Who is</td>
<td>Low</td>
</tr>
</tbody>
</table>

\(^{17}\) This measure is used by Piatek, et. al. (9) and by Siganos, et. al. (2)

\(^{18}\) Peer seen as a tuple [IP, port]

\(^{19}\) Autonomous System

\(^{20}\) File refers to the file being shared and hence peers distributing this file can be found in several swarms.
<table>
<thead>
<tr>
<th><strong>Protocol correctness</strong></th>
<th>Does the peer reacts as a normal client should or does the client doesn’t correctly implement the BitTorrent Protocol</th>
<th>No</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Upload speed</strong></td>
<td>Uploading speed of the peer</td>
<td>No</td>
<td>High</td>
</tr>
<tr>
<td><strong>Download speed</strong></td>
<td>Downloading speed of the peer</td>
<td>No</td>
<td>High</td>
</tr>
<tr>
<td><strong>Peer Passiveness</strong></td>
<td>Peer not asking or offering data</td>
<td>From selected peers</td>
<td>Medium</td>
</tr>
<tr>
<td><strong>Leeching Time</strong></td>
<td>Time during which the peer is reported in a swarm but is not seeding (“incomplete” bitfield)</td>
<td>No</td>
<td>Medium</td>
</tr>
<tr>
<td><strong>Seeding Time</strong></td>
<td>Time during which the peer is reported in a swarm and has a “complete” bitfield</td>
<td>No</td>
<td>Medium</td>
</tr>
<tr>
<td><strong>Number of Sessions</strong></td>
<td>Number of different Peer ID’s reported and/or number of connections received from the same Peer</td>
<td>From selected peers</td>
<td>Medium</td>
</tr>
</tbody>
</table>

Table 2. Peer Properties

**5.1 Common Peer Behavior**

Common peer behavior as defined in this project refers to the observed behavior that is followed by at least 99% of the peers based on the measured properties defined above. In the following subsections, common peer behavior will be defined based on three Indirect Monitoring properties (Popularity, IP Properties-Number of Ports and Total Connection Time).

**5.1.1 Popularity**

All of the observed popularity measures defined in Table 2 tend to follow an exponential distribution and as expected, the difference between the different measures is not high. Figures 8a and 8b below illustrate the distribution from Trace A and Trace B respectively.

21 Although it is possible to get a raw estimate from selected peers.

22 The presented results exclude the nodes that are reported by the trackers which result as an effect from the execution of the instrumented client.
It can be observed that for both traces 99% of the peers are found in 10 or less swarms and that 99% of the peers are reported in 5 or less files. This same observation can be done for the IPs, where 99% of the IPs are found in 10 or less swarms and are reported in 5 or less files.

5.1.2 Ports reported

One of the IP properties that can be easily measured from the traces is the number of ports gathered from the trackers for each IP. As depicted in Figure 9, over 99% of the IP addresses report the use of one or two ports, fact that is consistent across both of the traces gathered.

In addition to the number of ports reported per IP, it is of interest to analyze the ratio of files/ports per IP. As established in Figure 9, the use of only one port by the users appears as a very common behavior; however, according to the BitTorrent Protocol

Created with ReaSoft PDF Printer free trial.
Purchase at http://www.reasoft.com/
Specification (6) a user might use a single port to listen connections from multiple files.

Figure 10 illustrates the CDF of the files/ports ratio per IP, showing that 99% of the IP’s have a files/ports ratio of less than five, and that about 80% of the IP’s have a ratio of 1, i.e., one port per file.

![Figure 10a. CDF for Files to Ports Ratio per IP](image1)

![Figure 10b. CDF for Files to Ports Ratio per IP (Region)](image2)

5.1.3 Peer Connection Time

Although the establishment of the real peer connection time is a difficult task, specifically when this property’s estimation is based on the information returned by trackers where different aspects need to be taken into consideration, like the randomness of the set of peers returned, the freshness of the data in the tracker, among others, one approach for the estimation of the peer connection time is to take the first and last times a peer was reported during the time of the trace.

It is recommended when using this approach to perform the estimation over a long period of time to avoid the effect of noise in the data. This noise might come in the form of peers that were just leaving the swarm when the trace gathering was started, peers that just joined the swarm moments before the trace gathering was concluded, among others.

Figure 11 shows the CDF of peers’ connection time in hours. The time axis is presented as a percentage of the total trace duration, noticing that about 90% of the peers of Trace A are reported for a duration of 10% of the total trace duration and that about 80% of the peers of Trace A are reported only once.

---

23 Figure 10b is a close-up of the region shown as a dashed box in Figure 10a.
5.2 Summary of Common Peer Behavior

Based on the properties used in this section, which are estimated using the BitTorrent traces gathered, the following table summarizes a peer profile for Observed Common Peer Behavior:

<table>
<thead>
<tr>
<th>Peer Property</th>
<th>Common value range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Swarms per Peer and Number of Swarms per IP</td>
<td>[1, 10]</td>
</tr>
<tr>
<td>Number of Files per Peer and Number of Files per IP</td>
<td>[1, 5]</td>
</tr>
<tr>
<td>Number of Ports Reported per IP</td>
<td>[1, 2]</td>
</tr>
<tr>
<td>Files/Port ratio per IP</td>
<td>(0, 5)</td>
</tr>
<tr>
<td>Peer Connection Time (as a percentage of the total trace time)</td>
<td>[0%, 79%]</td>
</tr>
</tbody>
</table>

Table 3. Common Peer Behavior values
6. Deviant Peers and Monitors

Following the study of the observed common peer behavior, it is of interest to analyze the behavior of deviant peers based on the estimated properties defined above. In general, deviant peers can be classified as peers that present suspicious activity and peers that with a high level of confidence can be said to be monitoring the BitTorrent network, as shown in Figure 13.

![Diagram of Peer Behavior Classification]

Figure 13. Peer Behavior Classification

Monitoring peers are related to users that may be associated to law enforcement agencies (e.g. copyright protection), academic or industry researchers, or others whose real purpose is not to share/distribute files but to gather data from the BitTorrent network like details about peers distributing files, data about the files being distributed, etc.

Deviant peers are peers that although they present a deviant behavior it cannot be established with a high level of confidence whether they are monitors or not. These suspicious peers can be further classified as Justifiable Deviant Behavior, where the suspicious behavior can be explained possibly as part of a technical setting, for example, the use of a NAT by an ISP which would cause deviant levels of peer popularity.

In the same way, deviant behavior can also be classified as Non-Justifiable Deviant Behavior in which the peers present a deviant behavior but there is no clear explanation as to the reason of it. This may happen, for example, when the use of ports cannot clearly be associated with the behavior of a NAT and the whois information available for such peers is not enough.

---

24 The word suspicious is taken from blacklists (12) where some peers are marked as having suspicious behavior.
6.1 Deviant Behavior

From the traces gathered it is possible to detect deviant behavior using the properties established in Section 5. In the present project the classification of the peers is done by selecting those IPs that show at least one deviant property, based on the IP popularity (number of swarms and number of files), total connection time and port count properties.

<table>
<thead>
<tr>
<th>Number of Deviant Properties</th>
<th>Number of IPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11777</td>
</tr>
<tr>
<td>2</td>
<td>2245</td>
</tr>
<tr>
<td>3</td>
<td>1652</td>
</tr>
<tr>
<td>4</td>
<td>1139</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>16813</strong></td>
</tr>
</tbody>
</table>

Table 4. Number of Deviant IPs

As shown in Table 4 a total of 16,813 different IP addresses show at least one deviant measure. From this set of IP’s, 419 are reported in published blacklists (12). Information in blacklists is used for the purpose of crosschecking findings in regard of possible monitoring agents.

6.2 Detection of Monitors, Justifiable and Non-Justifiable Deviant Behavior

The process of detecting monitors is based on highly suspicious IP’s, i.e. IP’s that present a high number of deviant properties. It is important to clarify that the process of detecting IP’s that are likely to be monitoring agents is done manually to reduce the risk of false positives.

For the present project, the set of 1139 IP’s which present 4 deviant measures was initially checked, finding IP’s assigned to organizations like CheckTor, which admittedly monitors BitTorrent traffic, INRIA (French National Institute for Research in Computer Science and Control), Trend Micro Incorporated (Security Solutions Provider), among others.

Initially, the IP’s were checked against blacklists (12), which allowed an easy detection of a high number of highly popular IP’s related to the Emirates Telecommunication Corporation and to SingNet. In the case of the Emirates

---

25 The Classification process can be done using different properties. The properties used in the present study were chosen due to its availability from the traces gathered.
Telecommunication Corporation, the IP’s reported appear in 241 different swarms in average, while the IP’s from SingNet appear in 91 different swarms in average.

Both companies provide broadband residential services, so their behavior might be related to the use of Network Address Translation mechanisms. This argument is supported by the fact that none of the peers reported from these IP’s, where connection was attempted, accepted incoming connections.

When the remaining IP’s (i.e. filtering the Emirates Telecommunication Corporation IP’s and the SingNet IP’s) were sorted by the number of files being shared by each IP, a clear outlier which appeared in 1012 different swarms was observed. This IP address is assigned to an ISP in South Africa (Internet Solutions), and although its behavior is clearly suspicious, it can be justified as in the previous case, where the ISP maybe using NAT mechanisms.

This same behavior is likely to be followed by others ISP’s found in the list of suspicious IP’s, including Swedish company Ownit with a single IP reported in 547 swarms, United States company XMission with a single IP reported in 433 swarms, Philippines company GlobeNet with different IP’s in the suspicious IPs list including one reported in 371 swarms, UK company Vodafone with different IP’s including one reported in 25 different swarms, among others.

Although the possible justification of ISP’s using NAT’s seems reasonable due to the randomness of the ports being used, and to the impossibility to establish connections to the reported peers from these IP’s, it is also possible that these behavior might be due to a highly active user, or to other activities (e.g. monitoring); however the evidence gathered is not enough to establish this and the most likely reason based on the evidence seen is the use of NAT’s.

In addition to the possible NAT behavior by ISPs, which can be classified as Justifiable Deviant Behavior, examples of Non-Justifiable Deviant Behavior can also be found. Two IP’s, found in 274 and 247 swarms each, assigned to the Neutral Data Centers company show a very similar behavior to each other, i.e. both IPs are downloading 91 files in common, and the usage of ports is not concordant with the use of NAT mechanisms (e.g. same port is used to download several files).

In this case, although the activity can be associated with that of a monitoring agent, as the company provides Co-location Services and Bandwidth Solutions to its customers it is difficult to establish which company is generating the suspicious activity and to establish with certainty whether or not the company is a monitoring agent; hence, these IP’s are classified under Non-Justifiable Deviant Behavior.
Further to the ISP’s and other companies like Neutral Data Centers which provide bandwidth services to other organizations, it is possible to find suspicious behavior from IP’s which are assigned to organizations which are not involved in these kind of services. This is the case of the INRIA Sophia-Antipolis research centre, where one IP was reported in 140 different swarms, which is performing studies about BitTorrent, namely, “Understanding the Properties of the BitTorrent Overlay” (15).

Based on these facts and the behavior observed, including the fact that the IP address is using only one same port in 140 different swarms, which although allowed by the protocol specification it is a deviant behavior (99% of the peers are in a maximum of 10 swarms each), this IP address can be classified as a possible Monitoring Peer.

In addition to INRIA and as specified above, another possible monitoring agent was found using the mechanism described based in deviant measures. Three IPs assigned to the public, non-profit organization CheckTOR are found among the suspicious IP’s, where each IP appears on an average of 50 different swarms. CheckTOR admittedly monitors the BitTorrent network looking for viruses and other threats across the shared files.

Following the same process, an additional possible monitoring agent associated in the blacklists to the company MediaSentry\textsuperscript{26}, which was known as a copyright enforcement company (9), was found with an IP that is reported as downloading 7 different files. Although the amount of files is not highly deviant, i.e. only 2 files above the common behavior, the fact that it is blacklisted and that it presents this deviant property was of interest.

From this MediaSentry IP, five of the files being downloaded, although belonging to different torrent files, i.e. different info hashes for each of the five files, the content being distributed in each of the five files refers to the same content\textsuperscript{27}. Due to this behavior, it is possible to classify these IP, blacklisted as MediaSentry, as a possible monitoring agent.

6.3 Other Possible Monitoring Trends

During the development of the present project, different possibilities related to possible monitoring trends were proposed and some are due further study. The possible monitoring trends discussed in this section refer to probable mechanisms that are being used or that could be used by monitors to pretend being a regular peer, classified in this project as Smart Deviant Peers.

\textsuperscript{26} Recently acquired by ArtistDIRECT inc.; this is IP’s information is based on blacklists

\textsuperscript{27} E.g. each torrent file advertises “Harry Potter and the Half Blood Prince” with different variations in the name of the file, but it is clear that the file named in the five torrent files refers to the same media content.
Contracting Residential IPs:\footnote{Discussed also in (9)}

One of the ways for monitoring agents to perform their work avoiding easy detection is to perform their activities from Residential IPs which are managed by ISPs. Studies, including the one performed in this project, restrain the decision of classifying suspicious IPs assigned to residential service as monitors as there is not enough information available about the final user of the suspicious IP. Even more, by contracting different residential IPs with different ISPs it may be even possible to avoid detection of the IPs as deviant clients, by distributing the monitoring tasks across the different nodes and hence reducing the metrics discussed in section 5 for each IP.

Some disadvantages of using Residential IPs include the restrictions that might be imposed by the ISP in terms of firewall and security restrictions, and the possibility of needing a central web server to distribute the monitoring activities across the nodes.

Usage of Tor:

Especially useful for indirect monitoring, Tor provides a way to perform the monitoring and remain anonymous, as is the case of the present study. The indirect monitoring in this study was performed using Tor, which caused the metrics of several Tor nodes to be classified as deviant, but with no trace about the real origin of the suspicious behavior. This technique can be used to avoid or reduce the “risk of exposing monitoring agents” as defined by Piatek, et. al. (9).

One main disadvantage of using Tor is the final throughput of the system when compared to the throughput that can be achieved using direct connections as discussed in section 4.1.3.

Usage of the ‘stopped’ event:

One of the events that must be reported to the tracker by the peers is the stopped event when the client is shutting down (5). From the tests done after the data trace gathering, it was possible to observe that for the trackers tested, when a peer sent the stopped event it was immediately removed from the peers list of the tracker. Although tests must be performed in future work to establish if this behavior is consistent across all tracking servers, it can be or might be already being used by monitoring agents to avoid detection by sending a request to gather peers and immediately sending a stopped event to be removed from the list.
7. Conclusions

An analysis of peer behavior was presented, based on an empirical study from two BitTorrent traces gathered for a total of 7 and 2 days respectively. From the peer behavior analysis a common peer behavior profile was established and common values for some of these metrics were presented. It was shown that based on the establishment of these metrics deviant peer behavior can be established and an automatic peer classification can be performed, hence providing the possibility of using these metrics as heuristics to detect deviant clients that can complement those provided by Siganos, et. al. (2).

In addition, based on the peers’ classification made using Indirect Monitoring Peer Properties and by manually reviewing the list of suspicious peers, a set of possible monitoring agents was presented, showing that the detection of monitors is feasible using the heuristics provided and that as an extension of this, the BitTorrent networks are not only conformed by common clients, but also of clients with other purposes different to that of file sharing.

Future Work and Limitations

It is of interest to validate the findings of this project by performing studies that may crosscheck peers’ classification using different sets of heuristics and to gather traces of different timeframes (e.g. one week, one month, six months, etc.) to compare the results of the common peer behavior found. In addition, validation of the findings against a bigger set of torrents is desirable, including not only the most popular torrents, but also torrents with low popularity.

Studies related to the possibility to detect Smart Deviant Client are also encouraged; these studies might require heavy Direct Monitoring and the inclusion of additional properties to the set used in this project to detect deviant behavior.

Finally, the usage of the stopped event and the behavior of different trackers after receiving this event may clarify the possibility of it being used as a monitoring mechanism. Specifically, it is of interest to detect if most trackers remove the peer immediately from the swarm when this event is received.
8. Important Notes

The present project is intended for academic purposes only and no commercial profit is intended through the selling of the information/data gathered in the development of this project. Furthermore, the traces gathered as part of this study will be treated as private and will not be made publicly available or shared with third-parties.

Additionally, due to the sensitivity of the topic, the project, the BitTorrent clients used and the project related computer equipment will not be involved in the illegal sharing of data. Also, it is important to note that the author is not condoning unlawful sharing, and the only purpose is to provide a peer classification mechanism, and complement previous studies about the BitTorrent Protocol. Even more, findings about possible monitoring trends can be used to improve mechanisms to avoid detection of the monitoring clients, extending the challenges and directions given by Piatek, et. al. (9).
9. References


