Master Thesis

Revisiting the BitTorrent Ecosystem and Its Users

Steven Monteiro

Committee:
dr. Jeroen van der Ham-de Vos (DACS)
dr.ing. Florian Hahn (SCS)

Faculty of Electrical Engineering, Mathematics and Computer Science
University of Twente
The Netherlands
April 5, 2026
(Minor update: April 12, 2026)

Abstract

BitTorrent is arguably the most well-known peer-to-peer protocol and remains popular today. Despite this, academic research interest in the BitTorrent protocol has all but disappeared. We provide an updated look at the BitTorrent ecosystem using direct measurements of BitTorrent swarms. Additionally, we tailor our measurements to provide insight into a common use case for BitTorrent in the modern day: piracy. Our results show that the BitTorrent ecosystem has evolved, but there are no major upsets. Rather, it has evolved in a very predictable way, growing with technological changes such as IPv6 adoption and shifting legal landscapes.

Contents

1 Introduction **2**

2 Background **4**
2.1 The BitTorrent protocol . . . 4
2.1.1 Peer characteristics . . . 6
2.2 Torrent discovery sites . . . 6
2.3 Copyright infringement . . . 6
2.4 Anonymization . . . 7

3 Related work **9**
3.1 General real-world measurements . . . 9
3.2 Measurements of IPv6 traffic . . . 10
3.3 Measurements identifying peers . . . 10
3.4 Measurements targeted at copyright infringement . . . 11
3.5 Nontechnical works . . . 12
3.6 Other works . . . 12

4 Methodology **13**
4.1 Overview . . . 13
4.2 Detailed crawler architecture . . . 16

5 Results **18**
5.1 Country and network analysis . . . 18
5.2 IP and port analysis . . . 18
5.3 ISPs . . . 21
5.4 Comparison with Wolchok and Halderman . . . 21
5.4.1 Popularity distribution . . . 21
5.4.2 Torrent categories . . . 23
5.4.3 Torrent websites . . . 24
5.4.4 Top torrents . . . 25
5.4.5 IP classification . . . 25

6 Discussion **28**

7 Conclusion **29**
7.1 Future work . . . 29
7.2 Acknowledgements . . . 29

Bibliography **30**

Chapter 1

Introduction

The advent of the Internet has radically changed how we share information. In the early days of the Internet era, however, it was not yet clear which technologies were going to survive in the long term. One of the ideas that has stood the test of time is the concept peer-to-peer protocols, based on the idea that computers should request resources from each other rather than from specialized, centralized servers.

When discussing peer-to-peer, BitTorrent is the first name that comes to mind for many. The BitTorrent protocol has convincingly won the peer-to-peer popularity contest that took place in the early 2000s, with competitors such as eDonkey, Gnutella, Kazaa, and Napster having all but disappeared. Compared to these alternatives, BitTorrent has managed to sustain a strong community of so-called seeders and leechers, who are still engaging in high-volume data transfer today, efficiently sharing files with each other by distributing only pieces of those files at a time.

However, the popularity of BitTorrent is not uncontested. As alternative methods of file sharing emerged on the modern Internet, the spotlight shifted away from BitTorrent and towards cloud storage services, streaming platforms, Content Delivery Networks, and other client-server architectures [1]. This has not only resulted in a decrease in popularity of the BitTorrent protocol, but more relevantly also in a significant reduction in academic research interest in BitTorrent. A simple search for the term "BitTorrent" in the Scopus document index [2] reveals that research output, measured in the number of newly published documents per year, is at roughly 1/10th of what it was at its peak around 2010. This introduces an obvious preliminary question: Is fresh research into BitTorrent warranted, or is it a dying technology that is best left alone?

A recent report by Sandvine [3] convincingly answers this question: As of 2024, BitTorrent accounts for 4% of global upstream traffic, placing it above TikTok, Facebook, and Discord, some of the world's most recognizable tech brands. Insight into BitTorrent as it is used today would therefore be insight into a thriving ecosystem that is likely underexplored at the academic level. This provides a solid basis for our research: a comparison-based analysis between the characteristics of BitTorrent traffic that we can measure today and BitTorrent traffic that has been measured and published about in the past.

As mentioned, the peak of BitTorrent research volume occurred around 2010, and our research into related work indicates a stronger focus towards real-world traffic measurements from 2010 onwards. Before then, research mainly focused on performance analyses when it was not yet clear that BitTorrent would win the peer-to-peer protocol race. Therefore, we believe that it is reasonable to set a cut-off date of 2010 for our analysis, such that we only consider measurements in previous work from that year or later. Older works will be discussed in our chapter on related work. A generic surface-level analysis would however not do justice to the current state of the ecosystem: BitTorrent is well-known for attracting copyright infringers, as

we will elaborate on in further sections. To accommodate for this, we tailor our measurements to paint a more complete picture of the ecosystem. This is done by partially repeating the methodology of selected previous works, and by introducing a measurement that is novel to the best of our knowledge. This leads us to the following set of research questions:

RQ1 What are the changes in peers in BitTorrent swarms since 2010?

- **RQ1.1** What are the changes in commercial VPN and seedbox usage among peers in BitTorrent swarms since 2010?
- **RQ1.2** What are the changes in Internet and hosting provider usage among peers in BitTorrent swarms since 2010?

Following this set of research questions, it is important to note what exactly we intend to measure about BitTorrent peers. The oldest BitTorrent-related works especially focus on performance, a kind of analysis that heavily favors having a birds-eye view of the network with access to data such as NetFlow records. Not having this luxury, we instead turn to data that *identifies* the peer such as IP addresses. In our chapters on background and related work we will show that this type of data is well-suited to answer our research questions.

To further introduce the subject as well as the terms used in the research questions, we start by giving background information about the BitTorrent protocol. This is followed by a more detailed look at copyright infringement within BitTorrent. We will then discuss related work and its implications for the current research. We then describe the methodology by which we hope to answer the posed research questions. The results section gives insight into the measurements we have performed and explains the data as shown. We then continue to discuss these results in more detail by more explicitly comparing them to previous work and discussing their implications. We close with a conclusion that summarizes our findings with regard to the research questions and an indication of possible future work.

Chapter 2

Background

To introduce BitTorrent, it is only fair to start with the word itself. The word "BitTorrent" can refer to three separate but highly related phenomena. In its initial meaning during its original development, it refers to a peer-to-peer protocol. By extension, the name has been used to refer to the decentralized ecosystem that makes use of this protocol, in phrases such as "downloading movies from BitTorrent". Thirdly, BitTorrent is the name of the client software that served as the original implementation of the protocol and is still in use today, although the current version is no longer based on the original codebase [4].

Our research focuses on analysis of the ecosystem. It would however be improper to consider the ecosystem without first considering the protocol that it uses as its foundation. For that reason, we start by giving an overview of the protocol itself. We then continue with relevant aspects of the ecosystem. Note that we do not give any special attention to the client that carries the BitTorrent brand name: It is only one of many available today, such that filtering for it would more likely than not result in a study suffering from tunnel vision.

2.1 The BitTorrent protocol

BitTorrent [5, 6] is a peer-to-peer protocol that was initially developed in 2001 by Bram Cohen. Unlike the traditional client-server model, where a single server handles the distribution of files to all clients, BitTorrent allows clients (peers) to share files directly with each other. This distributed approach significantly reduces the strain on any single peer and simultaneously introduces redundancy in case of technical failures. Files are broken into small pieces, which can then be freely downloaded and uploaded between peers.

BitTorrent operates on groups of peers known as swarms. Each swarm is associated with one torrent, a user-created bundle of files and folders to be shared. The core idea is that each peer in a given swarm acts as both a downloader (for pieces it does not have yet) and an uploader (uploading pieces it does have to other peers that do not have those respective pieces yet). The protocol encourages participation: Each peer can request pieces at will, but peers that participate in uploading are prioritized, creating an incentive to give back rather than only taking away. This is also meant to ensure the overall health of swarms, given that a swarm cannot perform any useful activity if there are no peers willing to upload. Since many peers in a swarm may simultaneously have the same piece available, every peer is in principle replaceable, creating a native form of redundancy. When multiple peers do have the same piece available, they end up sharing the load of distributing that piece to other peers, accelerating the distribution process and making BitTorrent highly scalable as well.

It should be noted that the principle of giving back rather than only taking away is more than just technical. Within communities that make use of BitTorrent for file sharing, it is

considered poor form to refuse to give back, to the point where an IP address may be banned by its peers for having a too low upload-to-download ratio. Additionally, uploaders are known as seeders, while downloaders are known as leechers, carrying the negative connotation of a leech.

In order to download a file or set of files through BitTorrent, the user must first locate a torrent containing the desired content. BitTorrent does not provide a method to search torrent contents, so this step takes place through third-party services. These services are commonly torrent discovery sites, which we discuss in more detail later. Once the user has found a suitable torrent, the process is handed off to their torrent client, which is a peer. In order to participate in a torrent, the peer must locate the peers in the swarm for that torrent, a process known as peer discovery. This is a nontrivial problem: Since peers are not dedicated servers, there is often no established contact point at which they can be reached. There are three solutions to this problem in common use today: trackers, DHT, and PEX. Since these solutions are not mutually exclusive, they are often used at the same time.

Torrent trackers [6] Torrent trackers are specialized servers that coordinate swarms by providing a list of online peers for a given torrent. A newly joining peer can use this list of peers to establish contact and efficiently join the swarm. Given that trackers are specialized servers, they could be seen as centralized infrastructure, but with a far less crucial role than in a traditional client-server architecture: While trackers introduce peers to each other, peers that already know each other can connect directly without the help of a tracker. In order for peers to get started, a list of trackers is often included as part of a torrent's metadata when the user adds it to their torrent client. Peers announce their presence to the tracker when they join the swarm, as well as periodically during the session to confirm that they are still online. Trackers can be divided into public and private: Public trackers can be used by anyone knowing their address, whereas private trackers are subject to some kind of sign-up requirement which can vary heavily. Private trackers may also monitor their users to ensure they are contributing enough as defined by the tracker's rules.

Distributed Hash Table (DHT) [7] Although we discuss DHT here as part of the peer discovery process of the BitTorrent protocol, it is more properly considered a completely separate peer-to-peer network of which this is just one possible use case. Unlike BitTorrent, which can effortlessly share terabytes of data, DHT is most suitable for small amounts of data, but with the upside that the data is globally searchable by key. It is this searchability while still being decentralized that makes DHT uniquely suitable for peer discovery. DHT operates by storing key-value pairs in a distributed network of nodes, each responsible for a small part of the potential key space. In the context of Mainline DHT – the specific DHT network used for BitTorrent – the keys correspond to infohashes, a cryptographic hash of a torrent's metadata that serves as an identifier for the torrent, while the values correspond to lists of peers. To find peers for a torrent, the client iteratively queries the nodes in the DHT to find the node responsible for the infohash, getting closer with each query. Once the responsible node has been located, the peer obtains the list of existing peers from it and requests itself to be added to that list. This decentralized architecture consisting of many nodes eliminates the single point of failure associated with centralized trackers and therefore contributes to the resilience of BitTorrent in case trackers go offline. In line with resilience, clients that query the DHT are also expected to serve requests for others.

Peer Exchange (PEX) [8] Finally, PEX facilitates P2P communication by allowing peers to directly exchange information about other peers they are connected to. When a peer

joins a swarm, it can request a list of other peers from the peers it already knows about. The peer receiving the list can then share this information again if it itself receives a PEX request. Unlike the other solutions, PEX is subject to a bootstrapping problem, since a peer must already know about at least one other peer in order to perform a PEX request. As a result, PEX cannot be used as a standalone solution, but is used together with trackers and/or DHT to speed up the discovery process.

2.1.1 Peer characteristics

Within the context of this work, it is useful to note that peers tend to have certain identifying features and behaviors. The most obvious identification is the client field, equivalent to the user agent in web browsers. This field is sometimes used to filter peers to only acceptable clients, but this is not foolproof as the string can be trivially spoofed. Arguably the most important identification is the IP address of the peer. Together with the port, this holds a surprising amount of information about the peer. For example, it can be used to find the same peer across multiple torrents or to determine which country the peer is located in. Due to the age of much of the previous work, IPv6 is also a relevant statistic, since adoption of that protocol version has been steadily growing [9].

2.2 Torrent discovery sites

Torrent discovery sites, also known as torrent indexes, are websites that list torrents. They are intended to be used directly by humans and operate outside of the BitTorrent protocol. They however do not provide direct downloads of files. Instead, they provide torrent metadata that can be entered by the user into their torrent client. This takes the shape of torrent files, which are simply bundles of torrent metadata, and magnet links, the same thing but in a more compact and shareable URL form. The main purpose of torrent discovery sites is to provide the search functionality that is missing from the protocol by allowing users to search for torrents by their title (given by the uploader) or the names of the files and folders it contains. Some sites are focused on one particular type of content (e.g. movies), while others allow content from multiple categories. It is common for a torrent discovery sites to come with its own torrent tracker: This makes it easier for users of the site to find each other in BitTorrent swarms.

One cannot discuss torrent discovery sites without mentioning The Pirate Bay. The Pirate Bay [10] is arguably the most famous and also infamous torrent discovery site. It hosts one of the largest known repositories of torrents in a searchable format and does not restrict itself to any particular category of content. This notably includes piracy, from which the project gets its name. Intense legal scrutiny has resulted in jail time for its Swedish founders [11], who unsuccessfully argued that the website does not host any infringing content but merely links to its location in BitTorrent. The Pirate Bay originally hosted its own torrent tracker, but it has since been shut down [12], while the discovery site itself is active to this day, although it is unclear who its current operators are.

2.3 Copyright infringement

Any discussion about BitTorrent would be incomplete without mentioning illegitimate distribution of copyrighted material, more commonly known as piracy. Piracy and copyright infringement are arguably the most common associations made with the protocol, also in connection with the previously mentioned Pirate Bay. When applied to BitTorrent, piracy

simply consists of sharing copyrighted material through the protocol without the rightsholder's permission. BitTorrent is frequently used to share books, movies, music, and software, both legally and illegally. The main difference with traditional platforms is its decentralized nature, making it much more difficult to effectively take down infringing content. Due to this functionality, BitTorrent has received significant scrutiny from authorities and copyright holders who see the technology as facilitating copyright infringement. This is despite the fact that the protocol itself is not illegal: any illegality is purely in the actions taken by the user.

Note that "illegal" can be confusing in this context: Here, we only consider legal content that was redistributed in an illegal manner because no permission was obtained from the copyright holder, and therefore copyright infringement took place. This is clearly distinguishable from content of which the creation or possession is illegal, such as child sexual abuse material. Although this category of material is also known to be shared via BitTorrent [13], we do not discuss this further in this work and explicitly leave any analysis of such sharing open for future work.

Although some rightsholders see BitTorrent as a vehicle for piracy, the suitability of the protocol for copyright infringement appears to be mostly a coincidence: The very same concepts that give BitTorrent its resilience against failures for technical reasons give it resilience against failures for legal reasons. If any peer becomes unavailable because they were forced to stop participating in the distribution of copyrighted material, the pieces it offered can simply be downloaded from the remaining peers in the swarm. Moreover, the lack of centralization makes it difficult to monitor which content is being shared in the first place, since the content of torrents is not searchable through the protocol. The existence of torrents that contain only legally shareable content makes blanket action against BitTorrent piracy infeasible.

2.4 Anonymization

Although the decentralized nature of BitTorrent makes full takedowns difficult, copyright holders and other interested parties such as law enforcement are not helplessly watching as copyright infringement takes place over BitTorrent. The main tool to directly address infringement is the Digital Millennium Copyright Act (DMCA) in the United States, under which the copyright holder can send a notice of infringement to the Internet service provider (ISP) that facilitated the infringement. In the case of BitTorrent, this process involves locating infringing torrents and then watching the participants in the swarm. Once a particular peer is confirmed to have participated in its distribution, a notice can be sent to the ISP, or more generally to the owner of the peer's IP subnet. This is in fact the only possible place to send such a notice, since BitTorrent does not require any other identifying information to participate in a swarm. In case of a home connection, the owner of the IP address will be the user's home ISP, who is then intended to inform the user of the DMCA notice, require them to cease the infringement, and warn them of the consequences of piracy. The ISP may also be compelled by a court order to identify the entity who was using the IP address at the time of the infringement, allowing the copyright holder to pursue legal action directly.

This workflow has a flaw: The owner of the IP subnet must act on DMCA notices. While DMCA notices are routinely processed all across the world, they are only legally mandated in the United States. If the ISP does not act on DMCA notices or even responds with a notice that they cannot provide personal information, the user is de facto anonymized. The most straightforward case is that the user's home ISP does not have a strict copyright policy and does not care to inform the user. This mainly appears to be common outside of Western countries.

If a user's home ISP is not suitable, commercial no-log VPN services can provide a solution. These providers sell privacy as a service by allowing users to route traffic through their servers

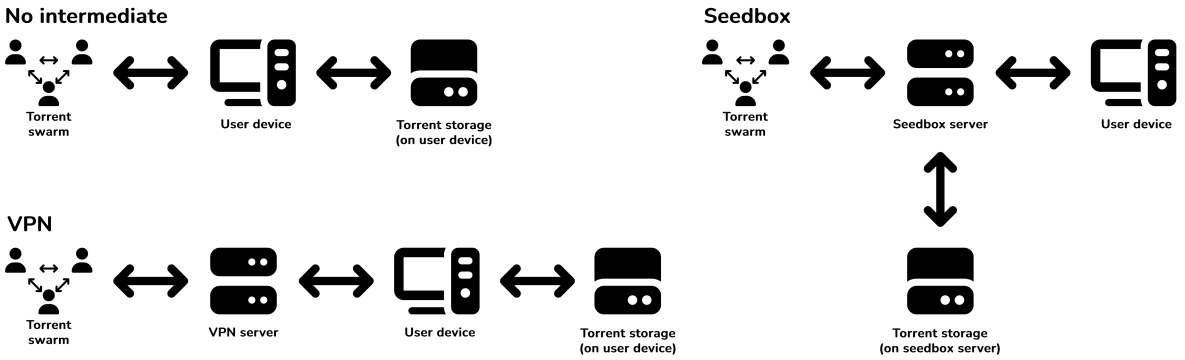


Figure 2.1: Schematic overview of connecting to BitTorrent swarms using different methods.

inside an encrypted tunnel while promising to not log traffic or traffic metadata. This hides the user's real IP address behind an IP address owned or rented by the VPN service itself. If a DMCA notice is received, the VPN service will (truthfully) inform the sender that they cannot identify the perpetrator since no logs are kept. Although this may seem like an easy legal loophole, VPN services have successfully argued in various countries that they are not ISPs and are not bound to logging requirements.

Savvier and larger-scale pirates may instead opt for a seedbox. Seedboxes are servers that exist for the explicit purpose of being active in torrent swarms, and especially to upload (seed) torrents long term. They aim for convenience with easy-to-use HTTP-based control panels rather than more technically demanding access methods like SSH. Seedboxes are offered by commercial providers, but can also be self-hosted using ready-made software packages such as swizzin [14]. Additionally, several popular torrent clients offer remotely accessible web panel interfaces as an alternative to their local GUIs. This means that although a seedbox is inherently a server, they may be found both in homes and in datacenters. However, the nature of a seedbox favors being active 24/7, having a stable Internet connection, and having dedicated dedicated storage space for torrents, meaning that most seedboxes at home are likely to be dedicated home servers rather than client devices. In terms of piracy, self-hosted seedboxes do not change the equation, since the user's home IP or VPN IP if applicable remain the same. Commercial seedbox providers on the other hand may collect enough billing information to act upon DMCA notices sent to their IP subnets. However, they generally actively choose not to, since they are often perfectly aware that their users are engaging in piracy but facilitate it.

These different methods of interacting with the BitTorrent ecosystem are shown schematically in Figure 2.1. Note that it is highly relevant which device is directly connected to the swarm, since it determines the IP address of the peer, the most important piece of identifying information.

Chapter 3

Related work

In this chapter, we review several works with relevance to the current research. It should be noted that, as we have previously mentioned, there has been a significant decline in academic research interest in BitTorrent today compared to 2010. For this reason, the majority of studies discussed are relatively old, namely one or even two decades, while the field of computer science normally moves much more quickly. It goes without saying that the measurements and analysis in these studies are likely not directly applicable to the BitTorrent ecosystem as it exists today. The methodologies should however be taken into consideration, since the basic BitTorrent specification has remained largely unchanged, with new features being added as optional rather than replacing old ones [15]. We split related work into several categories: general measurements of BitTorrent, measurements of IPv6 traffic in particular, measurements that aim to identify peers, measurements that focus on copyrighted works, and a brief highlight of assorted works.

3.1 General real-world measurements

Once the popularity of BitTorrent became apparent, studies of real-world measurements followed shortly after. Izal et al. [16] is one of the earliest long-term studies of activity within a torrent swarm. However, the applicability of this study is limited because only a single torrent was considered, and this torrent contained only legally redistributable content, namely a Linux distribution. This was soon corrected by broader follow-up studies [17, 18, 19, 20, 21], which each consider multiple torrents over a period of time. We in particular highlight the work by Pouwelse et al., performing measurements and analysis of BitTorrent swarms associated with suprnova.org, a popular torrent discovery site at the time that has long since been shut down [22]. Overall activity and availability were measured, as well as download performance under normal circumstances and in the period when a new popular torrent, such as a recently released movie, first becomes available. Their measurement approach consisted of separate scripts which respectively detect new torrents, contact trackers, and contact peers.

The ecosystem analysis of Zhang et al. [23] is one of the most well-known papers in the area of BitTorrent and significantly expanded the scope of real-world BitTorrent measurements. The researchers collected public English-language torrents from five popular torrent discovery sites at the time by crawling the respective websites. A particularly relevant result is that the study finds that "none of the major torrent sites on its own provides a complete picture of the Ecosystem", highlighting the need for diversified crawling. After querying trackers collected from torrents, trackers were ranked by the number of tracked peers and torrents. The torrents themselves were also categorized using both metadata contained within the torrent file and obtained from the torrent discovery sites. Lastly and most relevantly for the current study, a

country analysis is performed. Trackers, vulnerable to legal scrutiny due to their centralized nature, are mostly hosted in the United States and the Netherlands. Looking at the geographic distribution of all peers, Western countries lead the pack, with the United States having over twice as many peers as the number two country. However, the United Arab Emirates and Singapore are disproportionately represented when considering the number of Internet users in each country.

Hoßfeld et al. [24] measured BitTorrent swarms with various selection criteria. The authors note that the main purpose of their work is to provide input to the design of traffic optimization solutions in BitTorrent. For that reason, their analysis mainly focuses on swarm sizes and peer locality (peers that are close together from a networking perspective). Although this work seems only tangentially related to the instant research, we highlight it here because of its detailed analysis of peers by Autonomous System (AS), for instance plotting the number of distinct AS's seen per swarm.

3.2 Measurements of IPv6 traffic

In addition to the general real-world measurements detailed above, a few studies have measured specifically IPv6 in BitTorrent. However, research in this area appears to be sparse. Milescu and Bardac [25] measured the availability of IPv6 peers in the BitTorrent DHT by taking 120 popular torrents from The Pirate Bay and querying the DHT from IPv4 and from IPv6. Their analysis shows an IPv6-to-IPv4 ratio of around 1%, roughly in line with IPv6 adoption at that time. However, in the video category, the ratio was over 7%, which the authors attributed to the fact that video files are large, and therefore the IPv6 addresses have more time to be discovered. Lin et al. [26] used a sniffer on their university campus IPv6 network to measure IPv6 traffic and compare IPv6 TCP to UDP flows. However, the authors make the peculiar choice of not measuring IPv4 at all and thereby precluding potentially interesting comparisons, both within their work and with our measurements. Çiflikli et al. [27] measured IPv4 and IPv6 BitTorrent traffic and investigated the differences between them. However, although the analyzed traffic is BitTorrent, their work does not focus on BitTorrent properties but on general packet characteristics such as packet sizes and interarrival times, limiting the relevance to our research.

3.3 Measurements identifying peers

Le Blond et al. [28] identified multiple methods, deemed to be exploits by the authors, that together allow for efficient monitoring of BitTorrent users across swarms, compromising privacy. The first method involves monitoring a torrent discovery site for freshly added torrents and immediately joining the swarm. The first peer that already has all the pieces can then be assumed to be the creator. Additionally, if the discovery site displays a username for the uploader, the username can now be associated with an IP address. This approach works similarly without freshly added torrents if the discovery site allows grouping by uploader: The IP address that is present in most swarms likely belongs to the uploader. Lastly, the paper indicates that most trackers at the time supported a query that would return a full list of tracked torrents. Combined with the tracker's natural functionality of returning a subset of peers in the swarm for a given torrent, a large list of IP addresses and torrents they participated in can be obtained by periodically announcing oneself as a peer for each torrent.

In the same paper, Le Blond et al. proceeded to use their methods in real-world measurements to provide an analysis that is highly relevant to the current research. They successfully identified the creator's IP address for 70% of torrents newly added in the measurement period

and profiled each creator's contributions. A plurality of top creators used an IP address from server host OVH, possibly indicating a VPN or seedbox setup. The language of the torrent contents notably did not match the location of the servers. The authors then proceed with an IP address classification: By plotting the number of torrents the IP appears in against the number of distinct ports seen for the IP, IPs that are in use by multiple users due to NAT, VPNs, or similar, become clearly distinguishable from IPs that are in use by a single frequent user. Multiple categories of non-standard users were then identified: HTTP and SOCKS proxies were identified based on proxy lists, TOR exit nodes and VPNs were identified based on reverse DNS, other swarm monitors were identified based on lack of uploading, and frequent users were identified based on being willing to upload content across many torrents but with a low number of used ports. These identification methods are prone to false negatives, which the authors admit. They also identify HTTP and SOCKS proxies as belonging to anti-piracy groups.

Several studies have measured the DHTs used in BitTorrent [29, 30, 31]. We in particular highlight the study by Wolchok and Halderman, who significantly expanded the prevalent scope of measurement by showing that crawling the complete DHT used by BitTorrent is feasible on a single machine, and that BitTorrent user activity can be effectively monitored by using exclusively DHT and no trackers or torrent discovery sites. Using this method, the study found 7.9 million IP addresses downloading 1.5 million torrents in a period of 16 days, all observed from a single machine. The authors note that this result means it is feasible to quickly bootstrap a decentralized torrent index, possibly negating the need for centralized discovery sites, although copyright holders could use the same architecture to find infringing torrents faster. An identical port analysis to the one in Le Blond et al. was also performed, identifying several clear outlier IPs that belong to a VPN provider.

The work that is likely closest to ours is by Rossi et al. [32], giving a rare insight specifically into seedboxes. First, torrent creators were identified using a methodology similar to the one by Le Blond et al. discussed above, with additional correlation based on IP addresses that use multiple ports and multiple seemingly unrelated IP addresses participating in the same set of torrents. The authors concluded from their results that seedboxes may participate in swarms using multiple ports on the same IP, and that servers from completely unrelated providers may simultaneously be used by the same entity. The reverse DNS data of each peer was then manually inspected to detect seedbox providers and combined with manual web searches to identify and rank individual providers. Overall, seedboxes were found to participate in significantly more torrents than the average peer. The study additionally found that cheap providers with high general popularity were also popular for seedbox hosting.

Lareida and Stiller [33] performed one of the most recent BitTorrent measurement studies and mainly focused their analysis on identifying peer countries and AS's. The study first discussed country centrality as defined by multiple metrics and found that the United States is the most influential to the BitTorrent ecosystem "despite harsh measures against piracy", followed generally by English-speaking countries. Interestingly, the two most occurring AS's are both ISPs in the Philippines, while the number three is a hosting provider in the Netherlands. The authors accused the latter of hosting seedboxes and noted its relatively large presence compared to the country's population.

3.4 Measurements targeted at copyright infringement

Alcock and Nelson [34] performed measurements and analysis of New Zealand ISP traffic with the goal of characterizing sharing copyrighted content via P2P. Although their focus was on all forms of P2P traffic, their analysis explicitly covers BitTorrent, as well as several client-server protocols. The context of this study was the introduction of a new copyright law in New

Zealand that was intended to make it easier for copyright holders to target infringers. The study found a notable decrease in P2P traffic as the law went into effect, but a simultaneous increase in protocols used for tunneling, suggesting that some users may have switched to a VPN or seedbox. With regard to BitTorrent, the study found that the protocol continued to be the most significant bandwidth consumer after HTTP, even after the law had taken effect. Over 20% of the ISP's residential subscribers were recorded to have used some form of BitTorrent in each measurement.

A similar legal action in the Netherlands prompted Poort et al. [35] to analyze BitTorrent traffic: A court order was issued that required ISPs to block The Pirate Bay. After monitoring Dutch spoken or subtitled torrents over a longer period of time, the study found no notable decrease in downloading, although changes in traffic patterns were observed. The authors attributed this to psychological effects: The awareness effect, resulting from users being informed that illegal sharing is not appreciated by copyright holders, dominated any blocking effect where access to pirated content is made more difficult, although the awareness effect also wore off quickly and long-term effects on BitTorrent piracy were minimal.

A study that is close in nature to ours is Schmidt et al. [36]. The authors specifically considered illegal copies of content shared through BitTorrent, although their analysis leaned more heavily into the process of how an original work becomes a shareable torrent and who is responsible for that process. After this analysis, the paper moves to results with more relevance to our research, namely characterization of distributors. Trackers themselves are mostly hosted in the Netherlands and the United States, whereas frequent uploaders are often geolocated to France due to popular hosting provider OVH. The authors note the possibility of seedbox usage among this group, but do not investigate further.

3.5 Nontechnical works

In addition to the technical works described above, there are several relevant works of a nontechnical nature. In 2012, Larsson et al. [37] collaborated with The Pirate Bay to hold a survey on the use of VPNs among its users. 17.8% of the 75000 respondents used a VPN of some kind, and the percentage of VPN users is positively correlated with how frequently they upload. Danaher et al. [38] investigated how the blocking of piracy websites such as The Pirate Bay in the United Kingdom affects both subscriptions to legal services and circumvention behavior such as VPN usage. The authors argue that blocking is more effective when many piracy channels are closed at once, since it significantly increases the friction to continue pirating.

3.6 Other works

In addition to the previously discussed works, we briefly highlight the works by Qiu and Srikant [39], a highly cited paper providing one of the first detailed theoretical models of BitTorrent behavior; Piatek et al. [40], looking into how copyright holders monitor BitTorrent and showing that they are vulnerable to false positives; Siganos et al. [41], approaching peer identification from a different angle by focusing on generally identifying "deviant" peers such as swarm monitors; Bauer et al. [42], developing a framework to efficiently monitor very large BitTorrent swarms with a low false positive rate; and Wang and Kangasharju [43], showing that the BitTorrent DHT is vulnerable to sybil attacks allowing for extensive monitoring.

Chapter 4

Methodology

4.1 Overview

As we have discussed above, our study consists of a measurement of BitTorrent traffic. The first step in our methodology is therefore to obtain a list of torrents to monitor. Although a DHT monitor such as the one described in [30] would give the most comprehensive overview of the ecosystem, this would significantly expand the scope of the research, and as such make both the implementation of the monitor and the analysis of the data significantly more challenging. We instead take an approach similar to [23] by turning to the most popular torrent discovery sites as tracked by TorrentFreak [44] and taking a sample of their most popular torrents according to what is exposed by the website. For sites that have categories, we take samples from each category to ensure representativeness. For comparison purposes, we additionally sample the most popular Linux distribution torrents from Linuxtracker [45], which are all known to contain only legally redistributable content. Finally, we also consider Academic Torrents [46], a website for research datasets shared via BitTorrent. This results in the following list of measured torrent discovery sites:

1337X is a general-purpose torrent site, although most of its most popular torrents appear to be movies. It is notable for having an active community of uploaders as well as an active comment section for each torrent. It also features convenient pages listing trending torrents per category as well as overall, which we use for our crawler. Their trending page also lists the upload date for each torrent, allowing us to make a preliminary observation: Torrent popularity is significantly biased towards newer releases.

Academic Torrents is meant to host torrents containing scientific datasets for researchers and by researchers. Obtaining the list of most popular torrents is trivial, but its contents are noteworthy: A significant amount of the most seeded torrents are fragments of the publicly available database of Wikipedia articles in various languages. In addition to datasets, Academic Torrents features torrent downloads of online academic courses and lectures.

EZTV is a torrent discovery site that features exclusively TV shows. Unfortunately, the list of popular torrents we are able to use is short, instead more attention is given to brand new releases regardless of their popularity. This makes sense considering that TV shows are often released incrementally and therefore new episodes appear all the time.

linuxtracker is for Linux ISOs. Note that this website was not chosen for being particular popular on its own: Rather, it acts as an aggregator, giving us an easy entrance into the ecosystem of distributing Linux dsitributions by torrent. Unlike content from the other

websites, Linux ISOs can generally be redistributed legally, making them an excellent point of comparison. One way the torrents deviate from others is immediately obvious even without measurements: The most popular torrents are not the newest versions of their respective Linux distributions, whereas pirated content tends to strongly prefer freshness.

NYAA is dedicated entirely to anime and related content. The site often features new anime episodes shortly after they are released, but also manga (Japanese comics) and high-quality versions of Japanese songs. The main page of the website features an easy "sort by seeders" button that we use to obtain our sample.

The Pirate Bay is arguably the most well-known torrent discovery site in history and is still popular today. It is not restricted to any particular type of content. For our crawler, we take entries from the list of top torrents for each main category, as well as the overall top.

YTS exclusively focuses on Western movies and follows the idea of a now defunct site called YIFY, which prioritized movie releases with a small file size. Movie torrents are often first published on this website by torrent uploaders and are then known as YTS releases. YTS features multiple pages for trending torrents, although the exact inclusion conditions for each trending page are not clear. We take data from the pages called "24h YIFY Trending Movies", and search results for "ordered by featured", "ordered by seeds", "ordered by peers", and "ordered by downloads". It should be noted that there is a significant overlap in entries these categories, which is addressed by deduplication. Unlike most other torrent discovery websites, YTS lists multiple formats like 720p and 1080p of the same movie under the same listing, whereas they are separated on other platforms due to being different torrents. Some movies have as many as 6 formats, with the most common count being 2. When multiple formats are available, we add all available torrents to our crawler.

There are also several torrent discovery websites that we have chosen not to include in our measurements. Here, we go over each entry and explain our reasoning.

Anna's Archive has achieved notoriety [47] as a platform for copyright infringing content, mainly focusing on literature. However, its main distribution model is a traditional client-server architecture making use of many different server hosts. They do provide torrents, a very significant amount even, but their purpose is completely different than the torrents of the platforms we do include. Namely, they are not selected by content: Each torrent is essentially a completely arbitrary slice of files, and users are not intended to select torrents based on content at all. Instead, these torrents are exclusively intended to be downloaded by bulk storage servers that help keep the content alive by seeding, and that treat the content itself as essentially an opaque blob. Including Anna's Archive would therefore bias our dataset with torrents that function very differently from the others and whose contents are mainly distributed via client-server rather than peer-to-peer, meaning we would be missing a significant part of the puzzle regardless.

TorrentGalaxy is included in TorrentFreak's list of most popular torrent sites, but went offline in February of 2025 [48] before the start of our measurements and could therefore not be included for obvious reasons.

FitGirl Repacks, IGGGAMES, and Skidrow & Reloaded are three separate platforms that are grouped together here because of several common properties that make them unsuitable for inclusion. Firstly, each of them provides HTTP download links to the

content in addition to torrent files, making them not true torrent discovery websites. Secondly, none of the websites feature a list of popular torrents, or even a convenient list of torrents at all. There is no obvious way to obtain such a list other than by scraping the respective website, a task that is heavily out of scope for the current research. On top of that, FitGirl Repacks also uploads each of their releases to 1337X, a platform that we do include, ensuring that we do not completely miss out either way.

The actual list of torrents is obtained by visiting each torrent discovery website's page of most popular torrents and saving the magnet links. Obtaining a list from these websites only once would however not give a very accurate overview, since the BitTorrent ecosystem is alive and new content is regularly published. To address this, we manually repeat the collection procedure weekly and add the resulting magnet links to the crawler, creating a list that grows over time.

For the purpose of this study, we define popularity as raw swarm size, although website design often forces us to accept a certain ordering without knowing what numbers are used to calculate it. It should be noted that we trust the numbers given by the torrent websites, meaning that if torrent sites are engaging in suppression of certain torrents by artificially lowering the number of peers displayed in the interface, our measurement setup is not able not detect that. If numbers were artificially boosted, we could theoretically detect this based on measured swarm size, but this is an unreliable metric since the information in the interface could simply be stale, or the website may have access to a birds-eye view such as a tracker.

A similar note applies to categories: we copy the category as indicated by the website. The alternative would be to perform heuristics on the file names, but we argue that websites have little to gain from miscategorization, so it is a same assumption that categorization is rather accurate. Since we didn't define the categories ourselves, it's possible that some entries are categorized inconsistently. For example, anime is technically a subcategory of TV show, and a documentary can be a considered TV or movie. However, categories do generally match expectations, so that content with seasons is a TV shows and movies are feature-length, and anime has a distinct art style and usually originates from Japan. Software includes applications for all platforms, with the exception of video games, which are large enough to get their own category on most torrent sites, and we make this distinction in our work as well. The "other" category contains all kinds of data, such as books, audio books, and Western comics.

Having obtained a list of torrents, we turn to monitoring BitTorrent swarms. Our essential goal is to track as much data as possible. This approach has a major advantage: Unexpected patterns or correlations that are found in the data can be analyzed further without performing another round of data collection. A logical choice for our monitoring infrastructure would be one of the existing clients based on libtorrent [49], the most popular open-source implementation of the BitTorrent protocol. However, previous work [28] points out potential problems with this approach: Such clients are not designed for large-scale measurements and have relatively large computational overhead to implement end user conveniences. They may also be subject to blocking because they do not mitigate detection in any way, which the study also observed in practice.

For this reason, we first investigated reusing tools created by previous work that are able to perform large-scale measurements with minimal overhead. Unfortunately, code was often either unavailable or experimental to such a degree that it could not reasonably be used for a measurement that is not under constant supervision. Instead, we implement our own minimal client on top of the libtorrent library. This client functions as a crawler by entering each torrent swarm every 5 minutes and saving the list of peers it is able to find, then disconnecting again. No content is downloaded in this process: The client attempts to

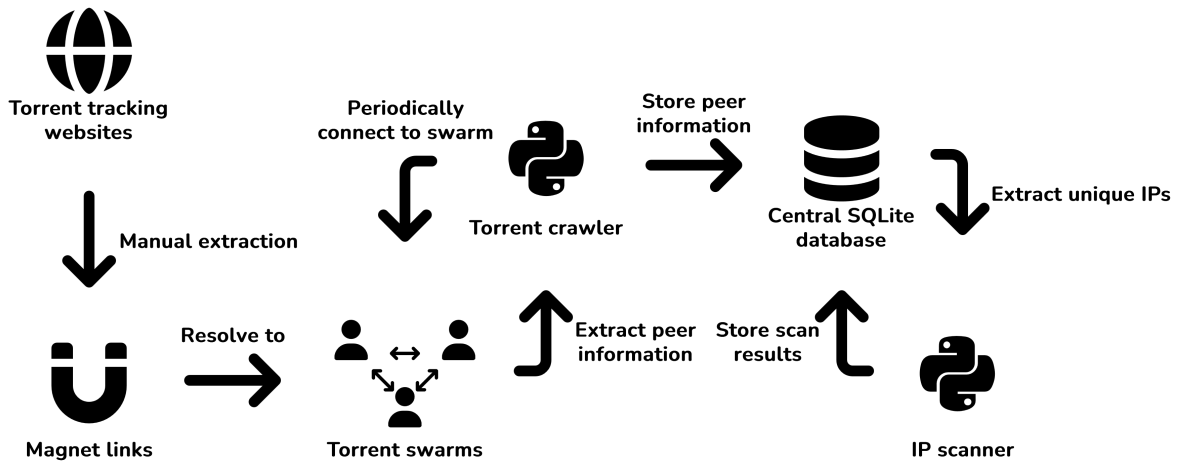


Figure 4.1: Technical overview of the architecture of the crawler.

locate peers, but never requests pieces from them. An earlier design instead attempted to simultaneously enter all swarms and keep track of their peers, but we found several practical problems with this approach. As noted by Le Blond [28], torrent trackers often limit the number of simultaneous connections from a single IP address to mitigate denial-of-service attacks, which would cause seemingly random tracker connection errors and missing peers. Additionally, keeping track of peers as a continuously changing list makes it very difficult to reason about the swarm as a whole, since determining the swarm at any given moment in time requires "replaying the journal" up to that point.

In addition to the regular swarm monitoring described above, we follow [28, 32] with a reverse DNS query of peer IP addresses, which may give away information about whether the IP is a VPN or seedbox or identify the provider. The reverse DNS query is performed only once for each unique IP address.

We add onto this with a measurement that is novel to our knowledge: We perform a port scan on each peer IP address to determine if the peer is hosting an HTTP control panel. Such control panels are expected to be common among seedboxes since they remove the need to access the server through more technical methods like SSH. Although these control panels may be firewalled like any other application, we deem this unlikely to be the default for seedbox providers since it would hinder access by their customers. In practice, this measurement involves making a full HTTPS request on port 443 and saving the response if applicable. Making the request on port 443 provides an information advantage compared to port 80, since TLS certificates contain hostname, possibly allowing us to determine the host or owner of the computer using the IP address. Saving the HTTP response provides a strong heuristic to separate IP addresses that are participating in torrents and also hosting an unrelated website from those that are hosting a website for the purpose of managing torrents on the machine. Examples of such websites are plain HTML pages stating "Seedbox Server", and login pages for seedbox software swizzin [14].

4.2 Detailed crawler architecture

In the preceding section, we have given a high-level overview of the crawler architecture. In this section, we discuss the architecture in more technical detail in order to highlight its features and design considerations. The measurements were performed by running the architecture described below on a dual-stack (IPv4 and IPv6-enabled) VPS located in the Netherlands.

Figure 4.1 shows a global overview of the crawler architecture. We start by obtaining magnet links from torrent websites in the manner discussed above. The magnet link allows libtorrent to connect to the swarm and discover peers using trackers, DHT, and PEX. The list of peers is stored in our central SQLite database. Using a central database simplifies analysis, since it allows tables to be cross-referenced and arbitrary results to be obtained via a single query.

The torrent crawler is a Python program that starts off with a list of magnet links that should be monitored. This list is static: Instead of allowing dynamic updates, the crawler is designed to function under unexpected restarts, which also makes it robust against operating system-level errors. The crawler makes use of an architecture with 100 runners. Each runner picks a single torrent from the global queue, joins the swarm, and then waits 60 seconds for peer discovery to complete. In this period, the connection is managed by libtorrent while the runner sleeps. After 60 seconds, we request the full list of currently connected peers from the libtorrent instance and save it to a SQLite database. We are then ready to disconnect from the swarm and readd the torrent to the global queue.

This architecture has one downside, namely that the expected interval between peer snapshots is dynamic. For any given swarm, it is low when there are few torrents to monitor and increases as the amount of monitored torrents increase. For example, at 2000 torrents and 100 runners, the expected interval is $2000/100 = 20$ minutes. This is an unfortunate but difficult to avoid side effect of making use of trackers for peer discovery, rather than a pure DHT architecture as envisioned by Wolchok and Halderman [30]. It should be noted that the values of 100 runners and 60 seconds are both heuristics. The limit of 100 comes from an observation by Le Blond et al. [28] that trackers blacklist peers that join over 100 torrents at once. Additionally, processing all torrents we know of at once is a very resource-intensive operation, since libtorrent is not a lightweight library. 60 seconds is a heuristic based on our own limited testing: It is entirely possible that this value is not optimal.

In our architecture, we offload the internals of BitTorrent to the libtorrent library as much as possible. The only setting of note that is modified is that we set the upload_mode flag on all torrents. In upload_mode, libtorrent only allows the client to upload pieces it already has, but since our client never starts out with any pieces of the torrents it monitors, we effectively prevent content from being downloaded while still fully participating in metadata exchange. We note that it is technically possible for other peers to detect that we participate in metadata exchange but never in downloading pieces. Therefore, we are not invisible in our monitor efforts.

The other component of the architecture is the IP scanner, which performs two tasks: reverse DNS querying and port scanning. Although reverse DNS is technically a feature in libtorrent, it is not well documented, so we favor the robustness of manual queries. Every hour, the script opens the central SQLite database and looks for IP addresses that have not yet been scanned. Each IP is then resolved using the Cloudflare public resolver, and the result written back to the database. Our use of Cloudflare here is coincidental and likely unimportant, since any performant DNS resolver should suffice.

The port scan is similar to the reverse DNS in setup, although slightly more involved. When an IP is found that has not yet been scanned, two requests are performed on port 443: The first retrieves and parses the TLS certificates sent by the server, the second performs a full HTTP GET request and saves the response. The results are then written back to the database. The fact that this is spread across two requests is a technical limitation caused by TLS operating at a different layer of the OSI model than HTTP. We also make note of a higher level limitation in this specific measurement: The port scan is IPv4-only since we were unable to obtain an IPv6 address that was cleared for port scanning by our network administrator.

Chapter 5

Results

Measurements were performed from 2025-08-12 to 2026-02-12. However, we warn against seeing this 6-month range as anything more than a sample of what took place in the BitTorrent ecosystem in these 6 months. There are several limiting factors: Since torrents were refreshed only weekly and manually, it is entirely possible that very popular torrents were released, grew in swarm size, and then shrunk in swarm size again in such a way that they were never captured. Additionally, the crawler was taken offline multiple times in this period for code changes and data extraction, meaning that there are known to be gaps in the data, sometimes significant ones.

In the rest of this chapter, we will highlight different aspects of the results of our measurements.

5.1 Country and network analysis

One of the most significant consequences of being able to obtain peer IP addresses is being able to group them by country by making use of an IP geolocation database, in this case IP2Location [50]. In Table 5.1, we show several statistics of BitTorrent peers. The United States tops the list with more than 2.5x the percentage of the number two, Canada. The IPv6 percentages are shown to vary quite a bit between countries, although it appears that all countries on the list are at least making partial progress with IPv6 implementation. The IPv6 percentages trail significantly behind Google's IPv6 adoption statistics [9], although this is no more than logical, since Google measures IPv6 support, whereas IPv6 support does not guarantee usage in practice with a dual-stack connection.

5.2 IP and port analysis

A particularly interesting method of classifying peer in BitTorrent swarms was first employed by Le Blond et al. [28] and shortly after by Wolchok and Halderman [30]. This method is notable for not using any metadata about the peer's IP addresses other than what can be directly observed: the IP itself, the port number, and the number of different swarms it is observed in. The plot itself consists of the number of unique torrents seen for a given IP address against the number of unique ports seen for that same IP address. This in theory allows peers to be divided into several categories simply by their positioning in the plot:

Residential peers are home users with a dedicated (though not necessarily static) IP address. They generally only use one port, since this is the default for all common BitTorrent clients, with a low number of unique torrents. Big downloaders are also in this category and can be recognized by the fact that they participate in more torrents than average.

Country	% of total	% IPv6	% Residential	% Datacenter	Count	% of total
United States	19.46	12.55	40.39	59.61	880,000	17.31
Canada	7.42	13.46	44.37	54.55	290,000	5.70
Netherlands	5.23	9.45	36.16	63.81	150,000	2.95
India	4.87	22.87	96.84	3.15	140,000	2.75
Australia	4.72	12.56	65.47	34.52	140,000	2.75
United Kingdom	3.83	12.30	67.52	32.46	320,000	6.29
Brazil	2.77	26.56	93.49	6.51	150,000	2.95
Switzerland	2.57	8.87	35.11	64.65	-	-
China	2.52	27.51	98.90	1.10	-	-
Sweden	2.37	7.06	46.97	52.94	-	-
France	2.28	15.83	55.56	44.43	230,000	4.52
Philippines	2.15	11.00	97.41	2.59	70,000	1.38
Germany	2.03	9.76	46.59	53.37	140,000	2.75
Israel	1.71	19.34	96.32	3.68	-	-
Spain	1.69	4.36	81.61	18.39	250,000	4.92
Portugal	1.39	11.02	94.03	5.97	60,000	1.18
Russia	1.39	6.15	94.29	5.71	80,000	1.57
South Africa	1.26	3.34	75.79	24.20	-	-
Romania	1.23	11.35	76.83	23.17	60,000	1.18
Singapore	1.17	8.44	43.91	56.07	80,000	1.57

Table 5.1: List of top 20 countries of torrents IPs in our measurements with percentages of different properties of the IP addresses within each country. Gray columns are from Zhang et al. [23].

Network	% of total	% IPv6	% Residential	% Datacenter
Datapacket (AS212238)	8.28	0.45	2.23	97.71
M247	4.55	6.72	0.02	99.98
PacketHub	4.02	0.00	0.49	99.51
Cogent	1.60	0.00	89.75	10.25
Clouvider	1.50	0.03	0.66	99.34
Comcast	1.46	38.16	100.00	0.00
Jio	1.45	37.17	100.00	0.00
Tzulo	1.43	14.48	0.00	100.00
Datapacket (AS60068)	1.41	20.74	0.00	99.93
31173 Services	1.39	10.37	0.00	100.00
AT&T	1.16	36.88	100.00	0.00
PLDT	1.05	17.57	100.00	0.00
GSL	0.98	0.00	1.09	98.91
Bharti Airtel	0.96	34.37	100.00	0.00
Aussie Broadband	0.81	18.91	100.00	0.00
DIGI Belgium	0.79	13.48	100.00	0.00
GTHost	0.76	0.00	5.22	94.78
Internet Utilities Europe and Asia	0.74	0.00	67.99	32.01
China Telecom	0.73	28.04	99.94	0.06
Verizon	0.72	24.47	100.00	0.00

Table 5.2: List of top networks torrent IPs belong to in our measurements with percentages of different properties of the IP addresses within each network.

Middleboxes can have many different kinds of behavior, but a common trend is for them to put multiple users behind a single external IP address. These IP addresses are expected to show up as a line in the plot, that is, the relation between torrents and ports is expected to be linear. This is because two distinct users cannot share a port in this setup, so as the number of ports increases, so does the number of users, and therefore the expected number of torrents for the IP.

There are several subcategories of middleboxes that may be present but are not always distinguishable. **SOCKS proxies** give users a convenient way to proxy traffic through an external server, but the protocol notably does not have a way to support incoming connections. **TOR exit nodes** can show up because the TOR network does not limit the kinds of traffic that can be sent through it, and therefore it can theoretically be used for BitTorrent, although this is not recommended by the project because of the load it puts on the network. Finally, **VPNs** allow users to connect via protocols such as OpenVPN and WireGuard and may also allow incoming connections through port forwarding.

Torrent monitors are servers that monitor the BitTorrent ecosystem. Although the scripts deployed during our measurements are also in this category, the majority of monitors belong to copyright holders and related organizations. These peers are distinguished by the fact that they participate in an unusually large number of swarms, often with very few unique ports (indicating only one BitTorrent application), and never actively upload material to peers, since this would be counter to their purpose.

To keep things clear, we present the resulting plot in four parts in Figure 5.1. The first part shows the density of IP addresses at each point in the plot. This is necessary because the amount of data points does not lend itself well to a graph that merely shows raw data points, although previous work did choose to present it as such. Since this plotting method

is intended to divide peers into categories, it is insightful to check the resulting plots if we filter by peers that have already been confirmed to be in one of the categories through other means, such as IP databases and reverse DNS lookups.

5.3 ISPs

When discussing who is participating in the BitTorrent ecosystem, it is not only the individual peers that are relevant, but also the networks they make use of. A high occurrence of a network in swarms for copyrighted material indicates a likely lackluster enforcement of copyright laws by the network provider. Furthermore, the network used by a peer is a strong indicator for the connection type: Residential ISPs typically restrict themselves to residential and business connections, while other networks are notorious for encouraging VPN companies to make use of their servers and IP addresses.

5.4 Comparison with Wolchok and Halderman

We attempt to compare our results to those in Wolchock and Halderman [30]. We believe this to be the most relevant comparison to earlier work because of their large-scale measurements on DHT. Other than their collection methodology of crawling DHT, as opposed to ours of taking torrents from torrent websites, there is a surprising amount of agreement between our measurements methods in terms of architecture. This is also visible in Figure 1 of their paper, which is similar to Figure 4.1 in our work. It goes without saying that other previous work must also be discussed; however, we believe that by starting with this comparison, a solid basis for analysis is created.

Note that Wolchok and Halderman include several results we intentionally do not replicate. One of the first results is a table for bootstrapping, related to catching up with the current state of DHT from a blank start. This is not relevant to our measurements since we do not crawl for torrents this way. The authors also provide a simple sorting of torrents into categories. This is possible because their measurement is neutral with regard to category, whereas we explicitly selected from categories to get an ecosystem overview. Therefore, we cannot compare how well each category is represented in the dataset. It is of course possible to provide statistics per category, which we do below. In addition, Wolchock and Halderman pick a single torrent and plot its popularity. This is entire within our realm of possibilities, but since we have no way to select a torrent with similar characteristics, there would be no basis for a comparison.

5.4.1 Popularity distribution

As visible in Figure 5.2, Wolchok and Halderman's graph supposedly follows the Zipf distribution, whereas ours is much flatter with a sharp dropoff. This however makes sense: their method of data collection gives a more balanced view of the torrent ecosystem, so it is only logical that it follows a law for natural distributions. Our sampling method takes only the most popular torrents, which are by definition biased towards being popular. Interestingly, one can interpret their graph to have the same features as ours if only the leftmost part of it is considered, where it is still mostly flat for the most popular torrents.

An interesting conclusion that can be drawn from our graph is that the most popular torrents have roughly equal popularity, as in that some series are not hugely more popular than others. It should be noted that this may still be the case over limited periods of time, such as that one torrent may have a much higher peak than another, since this is measuring

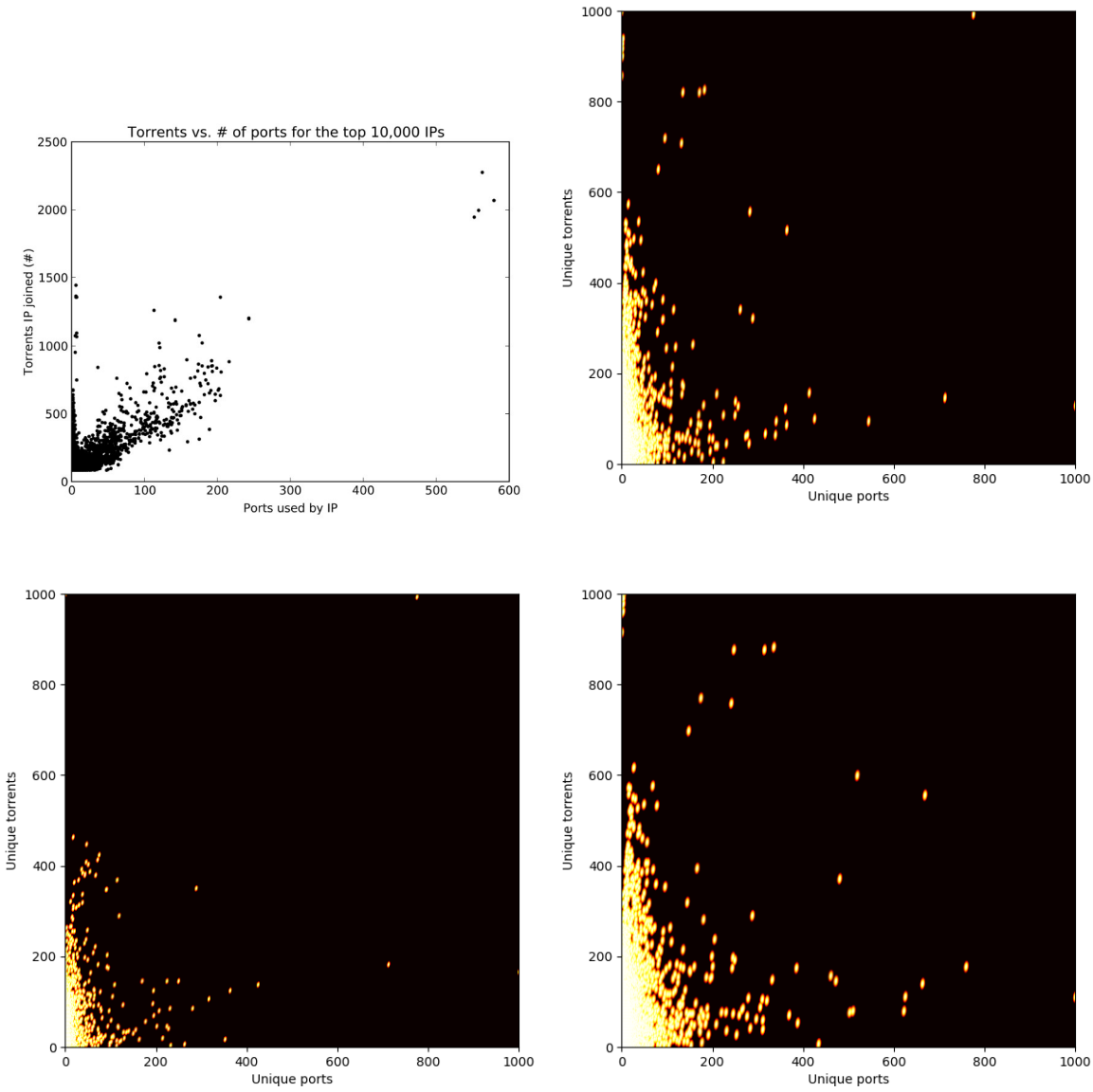


Figure 5.1: Plot by Wolchok (top left) and by us (others) of unique torrents against ports for the top 10000 IPs. Top right: all IPs. Bottom left: only residential IPs. Bottom right: only hosting IPs. See the body for explanation since what this shows is not very intuitive.

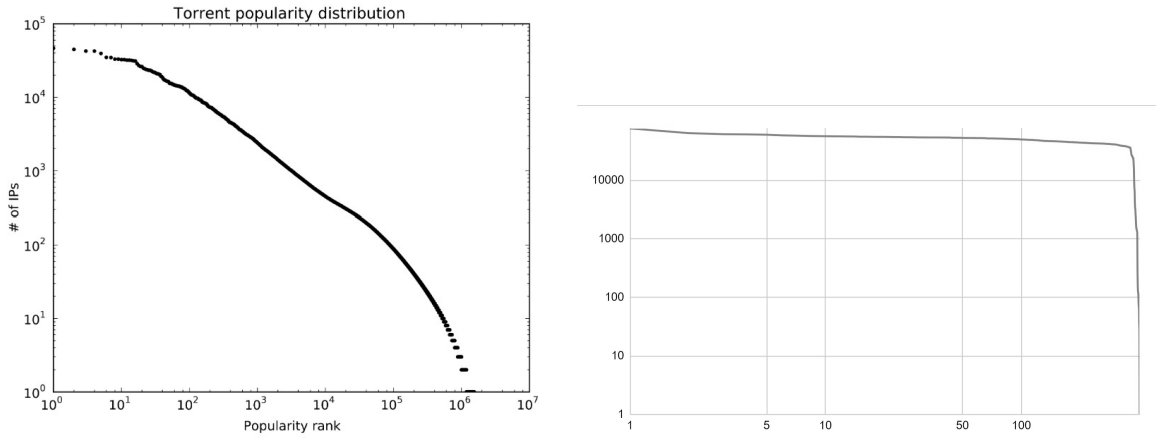


Figure 5.2: Left: Torrent popularity distribution copied directly from [30]. Right: Our version of the same graph. Double-logarithmic torrent popularity in number of unique IPs versus rank in our sample. Our result is significantly flatter with a steeper drop-off than Wolchok and Halderman. This can be explained by the fact that due to our sampling method, our dataset is more biased towards the most popular torrents.

the total amount of peers seen during the measurement period, as Wolchok and Halderman do too.

This raises the question of extracting the peak simultaneous peers: this would be very interesting, but we are limited by our measurement method, which does not guarantee that torrents are in our monitor at all when the peak occurs. If they are, we still cannot force the swarm to give us all peers, only whichever ones it wants, so we do not necessarily see the peak when it occurs, especially if this peak exceeds the limits of peer discovery packet sizes. This is a contributing factor to why we take repeated snapshots of the peer list rather than staying in the swarm permanently.

5.4.2 Torrent categories

Another question one might ask in response to the above is about categories, such as the relative popularity of software torrents compared to movie torrents. When answering this question, it is important to understand that we cannot measure popularity in terms of the amount of torrents, since we do not have a sufficient birds-eye view: we can only look at the popularity of the top torrents we know about. Notably, Wolchok and Halderman did have that kind of data but do not appear to have done that analysis. Our table can be seen in Figure 5.3.

The leftmost columns of this table require some explanation. We can define relative popularity in multiple ways. A simple method would be to consider the total number of unique IPs observed participating in each category. However, this would result in a bias, since the amount of torrents we monitor is not equal per category, and it seems logical to expect that more torrents results in a larger number of IP addresses. There are two obvious ways to address this: The first is to divide the raw number of IPs by the number of torrents we monitor in the category to normalize it ("normalized distinct peers"). The second is to simply calculate the average peers per magnet instead ("average peers per magnet"). The difference is in what happens when peers are in multiple torrents: for example, if the same 10 seeders are responsible for all activity in the entire category, that doesn't bode well for the category, while each individual torrent would be considered highly available. Therefore,

Category	Norm. distinct peers	Avg. peers per torrent	IPv6	Hosting
Anime	21.31	26.59	0.21	0.31
Books	12.81	16.95	0.12	0.28
Course	16.47	21.94	0.17	0.22
Dataset	17.74	22.93	0.11	0.34
Video games	23.42	25.55	0.20	0.11
Movies	23.95	28.99	0.08	0.20
Music	18.56	22.42	0.09	0.33
Pornography	20.49	27.46	0.12	0.21
Software	18.03	20.11	0.11	0.10
TV shows	28.47	32.53	0.09	0.26

Table 5.3: Torrent popularity by category.

Category	Norm. distinct peers	Avg. peers per torrent	IPv6	Hosting
1337X	18.41	23.36	0.15	0.19
Academic Torrents	18.00	23.20	0.13	0.28
EZTV	31.25	32.35	0.07	0.36
Linuxtracker	6.00	8.70	0.28	0.21
Nyaa	21.31	26.59	0.21	0.31
The Pirate Bay	23.93	27.83	0.10	0.18
YTS	22.99	28.17	0.08	0.22

Table 5.4: Torrent popularity by torrent website.

these are two different metrics that both give useful insight: The first gives an impression of the liveness of the category as a whole, while the second gives an impression of the torrents themselves.

Furthermore, we get insight from the delta between the two. Note that normalized distinct peers is always greater than or equal to average peers per magnet, since any given peer must appear in at least one torrent (otherwise we would not know about its existence). A large delta is an indication of many heavy torrenters in the category. Considering the pornography category as an example, we see that clients there are more likely to participate in multiple swarms than in the video game category.

The variation in IPv6 usage between categories is notable, but it is not possible to say what the cause of this is, since there are no IPv6-related variables that should affect some categories more than others. Hosting usage refers to the IP being a known hosting IP as opposed to residential. This is very interesting because of the two categories that dip significantly below the average: video games and software. Video games are of course a type of software, so it is interesting that they stand out together.

5.4.3 Torrent websites

Table 5.4 is the same as the categories we discussed above, but then with individual torrent websites. Here, Linuxtracker stands out for being much less popular than the others. This is logical, since Linux ISOs can also be directly downloaded and that is their most used distribution method, whereas copyright infringing content gains more from the resilience of BitTorrent. Linuxtracker also stands out for another reason: They have much higher IPv6 penetration than others. This, too, makes sense, since the audience of Linux ISOs is likely to

be much more technically inclined than the average torrent user. Interestingly, hosting usage is not far off the other values, so while it may be intuitive to conclude that Linux ISO peers do not bother to hide their source IPs, this is not immediately supported by the data.

5.4.4 Top torrents

Wolchok and Halderman do not do the category analysis we do above. Instead, they provide the table shown in Figure 5.5, which we follow with our own. As visible, this is a list of the top 10 torrents, and they are very similar in concept: all copyright infringing, (semi-)recent releases. This shows that whether a movie is known a "classic" is not a good indicator of its swarm activity, whereas freshness is. Interesting would be an analysis of whether fresh torrents of old movies are popular or not, but we suffice by noting that if such fresh torrents are significant, we would see those them in the list of popular torrents on torrent sharing websites and therefore they would already be included in our measurements.

Also note the specific entries of our top 10: For older TV series, it is the whole series being downloaded at once, whereas for new series we see individual episodes. This matches the behavior of Sonarr/Radarr-like programs that try to find torrents for users, giving a weak but possibility interesting signal towards their popularity. Unfortunately, this is hampered by two facts: We cannot distinguish them through metadata because they make use of standard torrent clients, and users manually searching likely exhibit the same behavior since it reduces friction. Therefore, we are most likely looking at natural behavior. This is also supported by Wolchok and Halderman's table, although only partially, since they don't have any full seasons of TV shows in their top 10. This is surprising considering their measurement was larger than ours and intuition says that complete seasons would be more popular overall than individual episodes barring special episodes.

5.4.5 IP classification

The IP classification graph is uniquely difficult to visualize when considering IP densities. The current visualization splits up the graph into categories with the hope of the "prongs" of the combined graph ending up in one graph each, which would give us good insight into what is causing each prong.

The graph is not entirety intuitive and requires some explanation: the central idea is that the amount of ports used by an IP address gives insight into what the IP is used for, because a single instance of a torrent client tends to assign a single port to itself. A very low number of ports with a high number of torrents therefore indicates a heavy user. But with a very high number of ports (¿100) while also being a datacenter IP, it is more likely a monitor, applications that are also measuring the BitTorrent ecosystem, except that their purpose is to send DMCA notices to seeders on behalf of copyright holders. A linear relationship indicates NAT, since we expect a relationship of the form $y = nx$ where n is the number of computers behind the NAT. Further to the right, we encounter VPNs and proxies, which appear similar to NAT from an external viewpoint, but are distinguishable since ISPs will not put 10000 clients behind a single NAT IPs, whereas commercial VPNs might do so over a longer period of time.

These points are also mentioned by Wolchok and Halderman while additionally pointing out 4 IPs all the way in the top right. These apparently belong to VPN provider Anchorfree, also known as Hotspot Shield, a provider anecdotally known for being free, not being very privacy-friendly, and being popular with school-aged children looking for ways to bypass school network restrictions. Anchorfree does not show up in our top 20 networks.

When comparing our graph to the one of Wolchok and Halderman, we can immediately notice that the shape is quite different. What is shared among them is that there is a dense

cluster in the bottom left corner, which represents IPs that only appear once or a few times in the measurements. We can see that both residential and hosting IPs are dense in this area, indicating "light" users. What is much more interesting is the prongs mentioned above, which clearly differ from Wolchok to our measurements and from residential to hosting. Note that some of the outlier points in the graphs appear to be present inconsistently among our three graphs. This is due to the top 10000 IPs being taken for each category separately rather than together.

First of all, it should of course be noted that the previous work had a higher peak number of torrents due to being a larger scale measurement. The group that follows the Y-axis upward is still visible in our data and represents either very heavy users or fellow monitoring software using very few clients to join many swarms. Interestingly, the main prong in the previous work does not appear to match up with ours, even considering the difference in axes. If anything, we observe much more outliers than Wolchok did. (Since Wolchok also took the top 10000 IPs, all outliers should be visible in their graph). Instead, we see multiple separate prongs. However, our hypothesis does not find strong support here: namely that the prongs are caused by different groups of IPs. Instead, it seems that except for the cluster around (300, 800), both residential and hosting IPs participate in the prongs. However, it is good to keep in mind that we are overall looking at outliers here, because the vast majority of IPs are in the dense region near the origin.

In general, the residential graph is more compressed towards the origin than the hosting one. This leads to the conclusion that when residential IPs are used, they are used less intensively than hosting IPs. This makes sense, considering a server is much more likely to be used by multiple users (in case of a VPN or proxy) or heavily used by one user (in case of a seedbox or monitoring effort).

Content	Type	Unique IPs
One Piece Episode 1136	TV series (anime)	77,980
EuroTrip	Movie	64,712
How I Met Your Mother Seasons 1-9	TV series	62,050
Dexter Seasons 1-8	TV series	61,592
One Piece Episode 1135	TV series (anime)	60,879
Sakamoto Days Episode 12	TV series (anime)	59,290
Sister Midnight	Movie	58,643
Clevatess Episode 3	TV series (anime)	57,963
Kaalidhar Laapata	Movie	57,538
Off the Grid	Movie	57,334

Content	Downloads
The Pacific, Part 9 (TV)	47,612
Iron Man (movie)	46,549
Alice in Wonderland (movie)	44,922
Lost, Ep. 16 (TV)	42,571
Dear John (movie)	42,562
The Back-up Plan (movie)	39,568
Lost, Ep. 13 (TV)	34,979
Desperate Housewives, Ep. 22 (TV)	34,783
The Moulin Rouge Soundtrack (music)	33,130
How I Met Your Mother, Ep. 22 (TV)	33,089

Table 5.5: Top 10 torrents in our measurements (top) and Wolchok's (bottom) copied from [30]. We find very similar results to Wolchok and Halderman: All top torrents are copyright infringing; 4 of them are individual episodes that were recently released as of the measurement period. Freshness appears to be significantly more important for swarm activity than long-term popularity.

Chapter 6

Discussion

In this chapter, we highlight several interesting features of our results and attempt to place these into context.

In Table 5.1, it was shown that the countries at the top of the list are mainly countries with a large online population, which of course makes logical sense sine a larger population has more opportunities to join BitTorrent swarms. The Netherlands seems to be an exception to this rule, having a much smaller population than the others. One possible explanation is that due to the excellent Internet infrastructure of the Netherlands, a disproportionate amount of datacenter providers are located here. This would explain why the datacenter percentage is the second highest out of all countries in this top 20. Another explanation is that there is a measurement bias caused by the fact that the measurement server was located in the Netherlands. For example, there could be peers that only respond to other peers from the same country, or peers that are only routable within the Netherlands.

In the same table, we notice that datacenter participation is significantly lower in some countries than in others. There are again two possible explanations here that are not mutually exclusive: These countries could simply be unfavorable for hosting due to regulations or costs, which is almost certainly why China has the lowest percentage, but another factor could be that hiding one's IP address is not needed as much in these countries, since ISPs there do not react strongly to copyright infringement. This is especially interesting since most of the countries that this applies to do not seem to be Western: India, Brazil, China, the Philippines, et cetera.

With regard to networks in Table 5.2, we observe that almost every network observed in the study can be placed into one of two categories: either predominantly residential or predominantly seedbox-oriented. This is consistent with the distinct roles played by residential ISPs and hosting providers in the global network ecosystem. Two networks, PacketHub and Internet Utilities Europe and Asia, give cause for special mentions. Based on manual investigation performed by us, these networks appear to have been specifically designed to host VPN IPs, and unlike other hosting providers, do not appear to onboard ordinary datacenter customers.

Chapter 7

Conclusion

We can conclude from our results that the BitTorrent ecosystem has both changed and stayed the same since 2010. For example, IPv6 has continued to grow and this pattern is also visible in BitTorrent. Statistics such as the top torrenting countries have been shuffled around. However, this also immediately reveals what has stayed the same: although the countries within the top 20 are shuffled, only 5 of the entries left or entered the top 20. This seems to be indicative of a broader pattern when looking at changes in the BitTorrent ecosystem, where there are no massive upsets, but rather very predictable changes that are caused by natural measurement variations as much as by the passing of time.

In general, the type of content BitTorrent users are consuming does not appear to be that different. Although copyright enforcement is much stricter now, the previous work clearly shows that BitTorrent has always been used for copyright infringement, and this has not noticeably changed in the past years. The actual content itself is of course different due to the time gap, but the trend continues to be that fairly recent releases are vastly more common than older ones. Overall, movies and TV shows apparently have always been the most popular categories of torrents and that continues to be the case.

7.1 Future work

This research has approached the BitTorrent ecosystem from the public perspective. However, there is another perspective worthy of consideration, namely the private tracker one. Private trackers are much more challenging to monitor than public trackers, since entry may require a personal invite, an application, or even a video interview. Unlike in public trackers, being pseudonymously monitored is fully expected on a private tracker, but there is an expectation of privacy that comes from being part of a closed group. Monitoring swarms in the way this study has done would likely be considered suspicious activity, since a copyright holder that has worked their way into the private tracker would exhibit the same behavior before issuing mass DMCA notices. However, private trackers would provide a very interesting insight into piracy behavior, since it is often stated on online forums that private trackers are perfectly safe to use with home IP addresses or with providers that are otherwise very strict on torrenting. Essentially, it would be an insight into what pirates do when they believe they are not being watched.

7.2 Acknowledgements

We are thankful to Jeroen van der Ham and Sanne Spuls for their boundless patience during the development of this thesis, which has been a long time in the making.

Bibliography

[1] Kenjiro Cho. Traffic shifting away from p2p file sharing to web services. Technical report, Internet Initiative Japan, 2010.

[2] Scopus. Scopus - document search results, 2025. https://www.scopus.com/results/results.uri?s=TITLE-ABS-KEY%28bittorrent%29.

[3] Sandvine. 2024 global internet phenomena report. Technical report, Sandvine, 2024.

[4] Ernesto Van der Sar. utorrent relaunched as official bittorrent client, 2007. https://torrentfreak.com/utorrent-relaunched-as-official-bittorrent-client/.

[5] Bram Cohen. Incentives build robustness in bittorrent, 2003. https://www.bittorrent.org/bittorrentecon.pdf.

[6] Bram Cohen. The bittorrent protocol specification, 2008. https://www.bittorrent.org/beps/bep_0003.html.

[7] Andrew Loewenstern and Arvid Norberg. Dht protocol, 2008. https://www.bittorrent.org/beps/bep_0005.html.

[8] The 8472. Peer exchange (pex), 2015. https://www.bittorrent.org/beps/bep_0011.html.

[9] Ipv6 – google, 2025. https://www.google.com/intl/en/ipv6/statistics.html.

[10] Download music, movies, games, software! the pirate bay - the galaxy's most resilient bittorrent site, 2025. https://thepiratebay.org/.

[11] Jemima Kiss. The pirate bay trial: guilty verdict, 2009. https://www.theguardian.com/technology/2009/apr/17/the-pirate-bay-trial-guilty-verdict.

[12] David Kravets. Pirate bay retires world's largest bittorrent tracker, 2009. https://www.wired.com/2009/11/pirate-bay-shutters-tracker/.

[13] George Bissias, Brian Levine, Marc Liberatore, Brian Lynn, Juston Moore, Hanna Wallach, and Janis Wolak. Characterization of contact offenders and child exploitation material trafficking on five peer-to-peer networks. *Child Abuse & Neglect*, 52:185–199, February 2016.

[14] swizzin community edition, 2025. https://swizzin.ltd/.

[15] David Harrison. Index of bittorrent enhancement proposals, 2008. https://www.bittorrent.org/beps/bep_0000.html.

[16] M. Izal, Guillaume Urvoy-Keller, Ernst W. Biersack, P. A. Felber, A. Al Hamra, and L. Garcés-Erice. Dissecting bittorrent: Five months in a torrent's lifetime. In Chadi Barakat and Ian Pratt, editors, *Passive and Active Network Measurement*, pages 1–11, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[17] Johan Pouwelse, Paweł Garbacki, Dick Epema, and Henk Sips. The bittorrent p2p file-sharing system: Measurements and analysis. In Miguel Castro and Robbert van Renesse, editors, *Peer-to-Peer Systems IV*, pages 205–216, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[18] Arnaud Legout, Guillaume Urvoy-Keller, and Pietro Michiardi. Understanding BitTorrent: An Experimental Perspective. Technical report, INRIA, 2005.

[19] Lei Guo, Songqing Chen, Zhen Xiao, Enhua Tan, Xiaoning Ding, and Xiaodong Zhang. Measurements, analysis, and modeling of BitTorrent-like systems. In *Internet Measurement Conference 2005 (IMC 05)*, Berkeley, CA, October 2005. USENIX Association.

[20] G. Neglia, G. Reina, H. Zhang, D. Towsley, A. Venkataramani, and J. Danaher. Availability in bittorrent systems. In *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pages 2216–2224, 2007.

[21] Lei Guo, Songqing Chen, Zhen Xiao, Enhua Tan, Xiaoning Ding, and Xiaodong Zhang. A performance study of bittorrent-like peer-to-peer systems. *IEEE Journal on Selected Areas in Communications*, 25(1):155–169, 2007.

[22] Ken Fisher. The death of suprnova.org, 2005. https://arstechnica.com/staff/2005/12/2153/.

[23] Chao Zhang, Prithula Dhungel, Di Wu, and Keith W. Ross. Unraveling the bittorrent ecosystem. *IEEE Transactions on Parallel and Distributed Systems*, 22(7):1164–1177, 2011.

[24] Tobias Hoßfeld, Frank Lehrieder, David Hock, Simon Oechsner, Zoran Despotovic, Wolfgang Kellerer, and Maximilian Michel. Characterization of bittorrent swarms and their distribution in the internet. *Computer Networks*, 55(5):1197–1215, 2011.

[25] George Milescu and Mircea Bardac. Ipv6 peer availability in bittorrent distributed hash table. In *2013 19th International Conference on Control Systems and Computer Science*, pages 532–536, 2013.

[26] Zhang Lin, Wang Hao, and Zhong Shibing. A measurement study on bittorrent traffic behaviors over ipv6. In *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, volume 3, pages 354–357, 2012.

[27] Cebrail Çiflikli, Ali Gezer, A. Tuncay Özşahin, and Öznur Özkasap. Bittorrent packet traffic features over ipv6 and ipv4. *Simulation Modelling Practice and Theory*, 18(9):1214–1224, 2010.

[28] Stevens Le Blond, Arnaud Legout, Fabrice Lefessant, Walid Dabbous, and Mohamed Ali Kaafar. Spying the world from your laptop: identifying and profiling content providers and big downloaders in bittorrent. In *Proceedings of the 3rd USENIX Conference on Large-Scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More*, LEET'10, page 4, USA, 2010. USENIX Association.

[29] Jarret Falkner, Michael Piatek, John P. John, Arvind Krishnamurthy, and Thomas Anderson. Profiling a million user dht. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, page 129–134, New York, NY, USA, 2007. Association for Computing Machinery.

[30] Scott Wolchok and J. Alex Halderman. Crawling BitTorrent DHTs for fun and profit. In *4th USENIX Workshop on Offensive Technologies (WOOT 10)*, Washington, DC, August 2010. USENIX Association.

[31] Liang Wang and Jussi Kangasharju. Measuring large-scale distributed systems: case of bittorrent mainline dht. In *IEEE P2P 2013 Proceedings*, pages 1–10, 2013.

[32] Dario Rossi, Guilhem Pujol, Xiao Wang, and Fabien Mathieu. Peeking through the bittorrent seedbox hosting ecosystem. In Alberto Dainotti, Anirban Mahanti, and Steve Uhlig, editors, *Traffic Monitoring and Analysis*, pages 115–126, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[33] Andri Lareida and Burkhard Stiller. Big torrent measurement: A country-, network-, and content-centric analysis of video sharing in bittorrent. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9, 2018.

[34] Shane Alcock and Richard Nelson. Measuring the impact of the copyright amendment act on new zealand residential dsl users. In *Proceedings of the 2012 Internet Measurement Conference*, IMC '12, page 551–558, New York, NY, USA, 2012. Association for Computing Machinery.

[35] Joost Poort, Jorna Leenheer, Jeroen van der Ham, and Cosmin Dumitru. Baywatch: Two approaches to measure the effects of blocking access to the pirate bay. *Telecommunications Policy*, 38(4):383–392, 2014.

[36] Adler Hoff Schmidt, Rodolfo Stoffel Antunes, Marinho Pilla Barcellos, and Luciano Paschoal Gaspary. Characterizing dissemination of illegal copies of content through monitoring of bittorrent networks. In *2012 IEEE Network Operations and Management Symposium*, pages 327–334, 2012.

[37] Ian Phau, Michael Lwin, Stefan Larsson, Måns Svensson, Marcin de Kaminski, Kari Rönkkö, and Johanna Alkan Olsson. Law, norms, piracy and online anonymity: Practices of de-identification in the global file sharing community. *Journal of Research in Interactive Marketing*, 6(4):260–280, 10 2012.

[38] Brett Danaher, Michael D. Smith, and Rahul Telang. The effect of piracy website blocking on consumer behavior. *SSRN Electronic Journal*, 2015.

[39] Dongyu Qiu and R. Srikant. Modeling and performance analysis of bittorrent-like peer-to-peer networks. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, page 367–378, New York, NY, USA, 2004. Association for Computing Machinery.

[40] Michael Piatek, Tadayoshi Kohno, and Arvind Krishnamurthy. Challenges and directions for monitoring p2p file sharing networks-or: why my printer received a dmca takedown notice. In *Proceedings of the 3rd Conference on Hot Topics in Security*, HOTSEC'08, USA, 2008. USENIX Association.

[41] Georgos Siganos, Josep M. Pujol, and Pablo Rodriguez. Monitoring the bittorrent monitors: A bird's eye view. In Sue B. Moon, Renata Teixeira, and Steve Uhlig, editors, *Passive and Active Network Measurement*, pages 175–184, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[42] Kevin Bauer, Damon McCoy, Dirk Grunwald, and Douglas Sicker. Bitstalker: Accurately and efficiently monitoring bittorrent traffic. In *2009 First IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 181–185, 2009.

[43] Liang Wang and Jussi Kangasharju. Real-world sybil attacks in bittorrent mainline dht. In *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 826–832, 2012.

[44] Top 10 most popular torrent sites 2025, 2025. https://torrentfreak.com/top-10-most-popular-torrent-sites-2025/.

[45] Linuxtracker .::. the premier linux bittorrent website, 2025. https://linuxtracker.org/.

[46] Academic torrents, 2025. https://academictorrents.com/.

[47] Ernesto Van der Sar. Major publishers sue anna's archive over 'staggering' copyright infringement, seek injunction, 2026. https://torrentfreak.com/major-publishers-sue-annas-archive-over-staggering-copyright-infringement-seek-injunction/.

[48] Ernesto Van der Sar. Torrentgalaxy drama continues with days of downtime (updates), 2025. https://torrentfreak.com/torrentgalaxy-drama-continues-with-days-of-downtime-250216/.

[49] Arvid Norberg. libtorrent, 2025. https://libtorrent.org/.

[50] Ip2location, 2025. https://www.ip2location.com/.