IJCSMS (International Journal of Computer Science & Management Studies) Vol. 22, Issue 01
Publishing Month: January 2016
An Indexed and Referred Journal with ISSN (Online): 2231-5268
www.ijcsms.com

A Review on Torrent & Torrent Poisoning over Internet

Pooja Balhara

Lecturer, All India Jat Heroes Memorial College, Rohtak, Haryana (India)
balhara.pooja2@gmail.com

Publishing Date: January 22, 2016

Abstract

Now a days, Torrents are widely used for downloading heavy files over internet the reason is being unlike other download methods, Torrent maximizes transfer speed by downloading the pieces of the files you want simultaneously from people who already have them. This process makes very large files, such as videos and television programs; games download much faster than is possible with other methods. In this paper, I am representing the basic and some popular details and activities after a deep review of torrent and torrent poisoning.

Keywords: *Torrent, bit torrent, torrent poisoning.*

1. Introduction to Torrent

A Torrent file is a computer file that contains metadata about files and folders to be distributed, and usually also a list of the network locations of trackers, which are computers that help participants in the system find each other and form efficient distribution groups called *swarms.* A torrent file does not contain the content to be distributed; it only contains information about those files, such as their names, sizes, folder structure, and cryptographic hash values for verifying file integrity. Depending on context, a torrent may be the torrent file or the referenced content.

Torrent files are normally named with the extension .torrent.

2. File Structure

A torrent file is a specially formatted binary file. It always contains a list of files and integrity metadata about all the pieces, and optionally contains a list of trackers.

A torrent file is a encoded dictionary with the following keys:

- Announce the URL of the tracker
- Info this maps to a dictionary whose keys are dependent on whether one or more files are being shared:
- Name suggested filename where the file is to be saved (if one file)/suggested directory name where the files are to be saved (if multiple files)
- Piece length number of bytes per piece.
- Pieces a hash list, i.e., a concatenation of each piece's SHA-1 hash. As SHA-1 returns a 160-bit hash, pieces will be a string whose length is a multiple of 160-bits. If the torrent contains multiple files, the pieces are formed by concatenating the files in the order they appear in the files dictionary (i.e. all pieces in the torrent are the full piece length except for the last piece, which may be shorter).
- length size of the file in bytes (only when one file is being shared)
- Files a list of dictionaries each corresponding to a file (only when multiple files are being shared). Each dictionary has the following keys:
- path a list of strings corresponding to subdirectory names, the last of which is the actual file name
- Length size of the file in bytes.

All strings must be UTF-8 encoded.

3. Bit Torrent

Bit Torrent is a communications protocol for the practice of peer-to-peer file sharing that is used to distribute large amounts of data over the Internet. Bit Torrent is one of the most common protocols for transferring large files, and peer-to-peer networks have been estimated to collectively account for approximately 43% to 70% of all Internet traffic (depending on geographical location) as of February 2009. In November 2004, Bit Torrent was responsible for 35% of all Internet traffic.[2] As of February 2013, Bit Torrent was responsible for 3.35% of all worldwide bandwidth, more than half of the 6% of total bandwidth dedicated to file sharing.

To send or receive files the user must have a Bit Torrent client; a computer program that implements the Bit Torrent protocol. Some popular Bit Torrent clients include μ Torrent, Xunlei, Transmission, q Bit torrent, Vuze, Deluge, and Bit Comet. Bit Torrent trackers provide a list of files available for transfer, and assist the client in transferring and reconstructing the files.

Programmer Bram Cohen, a former University at Buffalo student, designed the protocol in April 2001 and released the first available version on 2 July 2001 and the most recent version in 2013.

Bit Torrent clients are available for a variety of computing platforms and operating systems including an official client released by Bit torrent, Inc.

As of 2013, Bit Torrent has 15 27 million concurrent users at any time of the day. As of January 2012, Bit Torrent is utilized by 150 million active users (according to Bit Torrent, Inc.). Based on this figure, the total number of monthly Bit Torrent users can be estimated at more than a quarter of a billion. An extensive performance study of Bit Torrent protocols has been performed.

The Bit Torrent protocol can be used to reduce the server and network impact of distributing large files. Rather than downloading a file from a single source server, the Bit Torrent protocol allows users to join a "swarm" of hosts to upload to/download from each other simultaneously. The protocol is an alternative to the older single source, multiple mirror sources technique for distributing data, and can work effectively over networks with lower bandwidth. Using the Bit Torrent protocol, several basic computers, such as home computers, can replace large servers while efficiently distributing files to many recipients. This lower bandwidth usage also helps prevent large spikes in internet traffic in a given area, keeping internet speeds higher for all users in general, regardless of whether or not they use the Bit Torrent protocol.

A user who wants to upload a file first creates a small *torrent* descriptor file that they distribute by conventional means (web, email, etc.). They then make the file itself available through a Bit Torrent node acting as a *seed*. Those with the torrent descriptor file can give it to their own Bit Torrent nodes, which acting as *peers* or leechers download it by connecting to the seed and/or other peers (see diagram on the right).

Segmented file transfer implementation: the file being distributed is divided into segments called *pieces*. As each peer receives a new piece of the file it becomes a source (of that piece) for other peers, relieving the original seed from having to send that piece to every computer or user wishing a copy. With Bit Torrent, the task of distributing the file is shared by those who want it; it is entirely possible for the seed to send only a single copy of the file itself and eventually distribute to an unlimited number of peers.

Each piece is protected by a cryptographic hash contained in the torrent descriptor.[6] This ensures that any modification of the piece can be reliably detected, and thus prevents both accidental and malicious modifications of any of the pieces received at other nodes. If a node starts with an authentic copy of the torrent descriptor, it can verify the authenticity of the entire file it receives.

Pieces are typically downloaded non-sequentially and are rearranged into the correct order by the Bit Torrent Client, which monitors which pieces it needs, and which pieces it has and can upload to other peers. Pieces are of the same size throughout a single download (for example a 10 MB file may be transmitted as ten 1 MB pieces or as forty 256 KB pieces). Due to the nature of this approach, the download of any file can be halted at any time and be resumed at a later date, without the loss of previously downloaded information, which in turn makes Bit Torrent particularly useful in the transfer of larger files. This also enables the client to seek out readily available pieces and download them immediately, rather than halting the download and waiting for the next (and possibly unavailable) piece in line, which typically reduces the overall time of the download.

Once a peer has downloaded a file completely, it becomes an additional seed. This eventual transition from peers to seeders determines the overall "health" of the file (as determined by the number of times a file is available in its complete form).

The distributed nature of Bit Torrent can lead to a flood-like spreading of a file throughout many peer computer nodes. As more peers join the swarm, the likelihood of a completely successful download by any particular node increases. Relative to traditional Internet distribution schemes, these permits a significant reduction in the original distributor's hardware and bandwidth resource costs.

Distributed downloading protocols in general provide redundancy against system problems, reduces dependence on the original distributor and provides sources for the file which are generally transient and therefore harder to trace by those who would block distribution compared to the situation provided by limiting availability of the file to a fixed host machine (or even several).

One such example of Bit Torrent being used to reduce the distribution cost of file transmission is in the BOINC Client-Server system. If a BOINC distributed computing application needs to be updated (or merely sent to a user) it can do so with little impact on the BOINC Server.

4. Downloading Torrents and Sharing Files

Users find a torrent of interest, by browsing the web or by other means, download it, and open it with a Bit Torrent client. The client connects to the tracker(s) specified in the torrent file, from which it receives a list of peers currently transferring pieces of the file(s) specified in the torrent. The client connects to those peers to obtain the various pieces. If the swarm contains only the initial seeder, the client connects directly to it and begins to request pieces.

Clients incorporate mechanisms to optimize their download and upload rates; for example they download pieces in a random order to increase the opportunity to exchange data, which is only possible if two peers have different pieces of the file.

The effectiveness of this data exchange depends largely on the policies that clients use to determine to whom to send data. Clients may prefer to send data to peers that send data back to them (a tit for tat scheme), which encourages fair trading. But strict policies often result in suboptimal situations, such as when newly joined peers are unable to receive any data because they don't have any pieces yet to trade themselves or when two peers with a good connection between them do not exchange data simply because neither of them takes the initiative. To counter these effects, the official Bit Torrent client program uses a mechanism called "optimistic unchoking", whereby the client reserves a portion of its available bandwidth for sending pieces to random peers (not necessarily known good partners, so called preferred peers) in hopes of discovering even better partners and to ensure that newcomers get a chance to join the swarm.

Although swarming scales well to tolerate flash crowds for popular content, it is less useful for unpopular content. Peers arriving after the initial rush might find the content unavailable and need to wait for the arrival of a seed in order to complete their downloads. The seed arrival, in turn, may take long to happen (this is termed the seeder promotion problem). Since maintaining seeds for unpopular content entails high bandwidth and administrative costs, this runs counter to the goals of publishers that value Bit Torrent as a cheap alternative to a client-server approach. This occurs on a huge scale; measurements have shown that 38% of all new torrents become unavailable within the first month. A strategy adopted by many publishers which significantly increases availability of unpopular content consists of bundling multiple files in a single swarm. More sophisticated solutions have also been proposed; generally, these use cross-torrent mechanisms through which multiple torrents can cooperate to better share content.

Bit Torrent does not offer its users anonymity nor security. It is possible to obtain the IP addresses of all current and possibly previous participants in a swarm from the tracker. This may expose users with insecure systems to attacks. It may also expose users to the risk of being sued, if they are distributing files without permission from the copyright holder(s).

However, there are ways to promote anonymity; for example, the One Swarm project layers privacy-preserving sharing mechanisms on top of the original Bit Torrent protocol.

5. Torrent Poisoning

Torrent poisoning is intentionally sharing corrupt data or data with misleading file names using the Bit Torrent protocol. This practice of uploading fake torrents is sometimes carried out by anti-piracy organizations as an attempt to prevent the peer-to-peer (P2P) sharing of copyrighted content, and to gather the IP addresses of downloader.

Methods:

a) Decoy Insertion

Decoy insertion also known as content poisoning is one of the most popular method by which corrupted versions of a particular file are inserted into the network. This deters users from finding an uncorrupted version and also increases distribution of the corrupted file. A malicious user pollutes the file by converting it into another format that is indistinguishable from uncorrupted files (e.g. it may have similar or same metadata). In order to entice users to download the decoys, malicious users may make the corrupted file available via high bandwidth connections. This method consumes a large amount of computing resources since the malicious server must respond to a large quantity of requests. As a result, queries return principally corrupted copies such as a blank file or executable files infected with a virus.

b) Index Poisoning

In this method the index of the files are manipulated or altered by the malicious users. The index allows users to locate the IP addresses of desired content. Therefore making it difficult to locate the file to peers. The attacker inserts a large amount of invalid information into the index to prevent users from finding the correct resource. Invalid information could include random content identifiers or fake IP addresses and port numbers. When a user attempts to download the corrupted content, the server will fail to establish a connection due to the large volume of invalid information. Users will then waste time trying to establish a connection with bogus users thus increasing the average time it takes to download the file. The index poisoning attack requires less bandwidth and server resources than decoy insertion. Furthermore, the attacker does not have to transfer files nor respond to requests. For this reason, index poisoning requires less effort than other methods of attack.

c) Spoofing

Some companies that disrupt P2P file sharing on behalf of content providers create their own software in order to launch attacks. Media Defender has written their own program which directs users to non-existent locations via bogus search results. As users typically select one of the top five search results only, this method requires users to persevere beyond their initial failed attempts to locate the desired file. The idea is that many users will simply give up their search because of frustration.

d) Interdiction

This method of attack prevents distributors from serving users and thus slows P2P file sharing. The attacker s servers constantly connect to the desired file, which floods the provider s upstream bandwidth and prevents other users from downloading the file.

e) Selective Content Poisoning

Selective content poisoning (also known as proactive or discriminatory content poisoning) attempts to detect pirates while allowing legitimate users to continue to enjoy the service provided by an open P2P network. The protocol identifies a peer with its endpoint address while the file index format is changed to incorporate a digital signature. A peer authentication protocol can then establish the legitimacy of a peer when they download and upload files. Using identity based signatures, the system enables each peer to identify pirates without the need for communication with a central authority. The protocol then sends poisoned chunks to detected pirates requesting a copyright protected file only. If all legitimate users simply deny download requests from known pirates, pirates could usually accumulate clean chunks from colluders (paid peers who share content with others without authorization). However, this method of content poisoning forces pirates to discard even clean chunks, prolonging their download time.

f) Eclipse Attack

The eclipse attack also known as routing-table poisoning instead of poisoning the network, targets requesting peers directly. In this attack, the attacker takes over the peer s routing table so that they are unable to communicate with any other peer except the attacker. As the attacker replicates the whole network for the targeted peer, they can manipulate them in a number of ways. For example, the attacker can specify which search results are returned. The attacker can also modify file comments. The peer s requests can also be directed back into the network by the attacker and can also be modified. It also checks data randomly for any errors found in that.

g) Uncooperative-Peer Attack

In this attack, the attacker joins the targeted swarm and establishes connections with many peers. However, the attacker never provides any chunks (authentic or otherwise) to the peers. A common version of this attack is the "chatty peer" attack. The attacker establishes connection with targeted peers via the required handshake message, followed by message advertising that they have a number of available chunks. Not only does the attacker never provide any chunks, they also repeatedly resend the handshake and message. These attacks prevent downloads as, essentially, the peer wastes time dealing with the attacker, instead of downloading chunks from others.

6. Counter Measures

The methods of attack described above are not particularly effective on their own, as for each measure effective countermeasures have evolved. These measures must be combined in order to have a significant impact on illegal peer-to-peer file sharing using Bit Torrent protocols and Torrent files.

- Bit Torrent is highly resistant to content poisoning (as opposed to index poisoning), as it is able to verify individual file chunks. Overall, Bit Torrent is one of the most resistant P2P file sharing methods to poisoning.

- By Torrent users being members of Private Tracker websites (where one has to be a member of the Torrent tracker website) -- poisoned torrents can be quickly labeled and deleted and the person responsible can be banned from the site(s).

- Public torrent tracker sites have enabled the ability to report if a torrent has been poisoned (or is fake or malicious in any way). Thus torrent files shared by public trackers can have similar levels of quality assurance as Private Tracker websites.

- Tracker technology as well as Bit Torrent client programs have improved over time, and many kinds of spoofing that were possible in the past are no longer possible.

- Bit Torrent used to exclusively be a TCP-IP protocol, but this is no longer true. Use of UDP, with the uTP protocol has made TCP Man in the Middle attacks more difficult to nearly impossible.

- Public or Private tracker websites have selectively switched over to using SHTTP for the distribution of their web text and image content. By using SHTTP for the website content (versus tracker communications) many poisoning techniques are rendered impossible.

7. Conclusion

Through this paper, I want to state that Torrent is a very beneficial way to upload and download the heavy files over internet, but, its misuse cannot be neglected. Although it is clear that the torrent software are not illegal as they are very useful in information and data sharing but pirating the copyrighted materials over these networks are illegal. However, I am keeping it as a topic for open research. I hope the upcoming researchers could put some more light on this emerging topic for research.

References

[1] Cuevas, R. et al. (2010) Is Content Publishing in Bit Torrent Altruistic or Profit-Driven? Proceedings of the 6th International Conference on emerging Networking Experiments and Technologies (ACM CoNEXT 2010). Philadelphia, USA. 30 November - 3 December 2010.

[2] Locher, T. et al. (2010) Poisoning the Kad Network. In: Kant, K. et al (eds). Distributed Computing and Networking. Heidelberg: Springer. pp. 195-206.

[3] Kong, J. et al (2010). A Study of Pollution on Bittorrent. In: The 2nd International Conference on Computer and Automation Engineering. Singapore, 26 28 February 2010. New Jersey: IEEE. pp. 118-122.

[4] Kong, J. et al (2010) The Evaluation of Index Poisoning in Bit Torrent. In: D. Wen. et al (eds). Proceedings of the Second International Conference on Communication Software and Networks. Singapore. 26 28 February 2010. New Jersey: IEEE. pp. 382-386.

[5] Santos et al (2010). Choking Polluters in Bit torrent File Sharing Communities. Network Operations and Management Symposium (NOMS) 2010. Osaka, 19 23 April 2010. New Jersey: IEEE. pp. 559-566.

[6] Luo et al (2009). An Effective Early Warning Scheme against Pollution Dissemination for Bit torrent. In: Global Telecommunications Conference, 2009. Honolulu, 30 November 4 December. New Jersey: IEEE. pp. 1 -7.

[7] Lou, X. and Hwang, K. (2009) Collusive Piracy Prevention in P2P Content Delivery Networks. IEEE Transactions on Computers. 58 (7) pp. 970-983.

[8] Von Lohmann, F. (2008) A Better Way Forward: Voluntary Collective Licensing of Music File Sharing. Electronic Frontier Foundation. Retrieved 2011-04-22.

[9] Lou, X., Hwang, K. and Zhou,R. (2007) Integrated Copyright Protection in Peer-to-Peer Networks. In: 27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07). Toronto, Canada. 22 29 June 2007. p. 28

[10] Anderson, N. (2007). Peer-to-peer poisoners: A tour of Media Defender Ars Technica. Retrieved 2011-03-30.