INFO HASH TORRENT SEARCHING TECHNIQUE

[1] SURAJIT KARMAKAR, [2] POOJA MEHTA, [3] VAIBHAVI SHAH, [4] HARDIK SOMAIYA

Department Of Computer Engineering

[1]K. J. Somaiya Institute of Engineering and Information Technology, Sion, Mumbai.

[2]Shah and Anchor Kutchhi Engineering College, Chembur, Mumbai.

[3, 4]K. J. Somaiya College of Engineering, Vidyavihar, Mumbai.

surajit.karmakar03051994@yahoo.in, poojamehta.20893@gmail.com, vaibhavishah93@gmail.com, hardik.somaiya101093@yahoo.in

ABSTRACT: *In this paper, we look closely at the BitTorrent P2P protocol. We extract problems that have already been studied from the protocol and discuss those problems. We propose a system for efficient searching which indexes torrents from multiple sources so that the users can have access to a large number of torrents from a single source.*

KEYWORDS: ***Peer-to-Peer, BitTorrent, SHA-1, hash, database, caching.***

1. INTRODUCTION

Peer-to-peer networking, often referred to as P2P, is perhaps one of the most useful and yet misunderstood technologies emerging in recent years. When people think of P2P they usually think of one thing: sharing music files, often illegally. This is because file-sharing applications such as BitTorrent have risen in popularity at a staggering rate and these applications use P2P technology to work. Although P2P is used in file-sharing applications, that doesn't mean it doesn't have other applications. Indeed, as you see in this paper, P2P can be used for a vast array of applications, and is becoming more and more important in the interconnected world in which we live. The two protocols of P2P networks are:

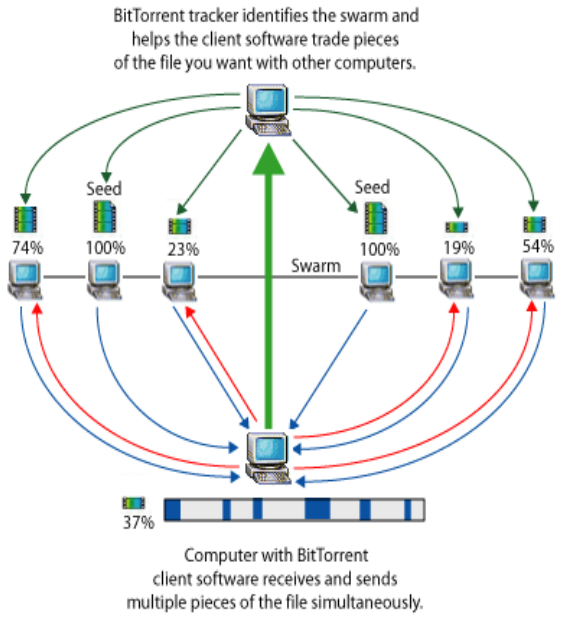
1. Direct Connect Protocol
2. BitTorrent protocol

Direct connect clients connect to a central hub and can download files directly from one another. Hubs feature a list of clients or users connected to them. Users can search for files and download them from other clients, as well as chat with other users. It is a text-based computer protocol, in which commands and their information are sent in clear text. As clients connect to a central source of distribution (the hub) of information, the hub is required to have a substantial amount of upload bandwidth available.

The biggest disadvantage is that while downloading from public hubs, although the receiver might have a higher bandwidth connection they will be limited to a lower bandwidth because of the lower bandwidth of the sender leading to waste of time and bandwidth. To share any file the sender must be online and while in offline phase, transmission of files in not possible with Direct Connect (DC).

The BitTorrent protocol is peer-to-peer in nature, its innovative approach in the beginning, was due to not be centered about the creation a real distributed network but around the specific shared resources, in this case files, preferably large files, as users connect to each other directly to send and receive portions of a large file from other peers who have also downloaded either the file or parts of the it. These pieces are then reassembled into the full file. Each downloader reports to all of its peers what pieces it has. To verify data integrity, the *SHA1* hashes of all the pieces are included in the .torrent file, and peers don't report that they have a piece until they've checked the hash. Since the users are downloading from each other and not from one central server, the bandwidth load of downloading large files is divided between the many sources that the user is downloading from. This decreases the bandwidth cost for people hosting large files, and increases the download speeds for the people downloading large files, because the protocol makes use of the upstream bandwidth of every downloader to increase the effectiveness of the distribution as a whole, and to gain advantage on the part of the downloader. However, there is a central server (called a *tracker*) which coordinates the action of all such peers. The tracker only manages connections, it does not have any knowledge of the contents of the files

being distributed, and therefore a large number of users can be supported with relatively limited tracker bandwidth. By reducing dependency on a centralized tracker, *PEX* increases the speed, efficiency, and robustness of the BitTorrent protocol. Within BitTorrent, a *torrent file* is a computer file that contains metadata about the files to be shared and about the tracker, the computer that coordinates the file distribution. A *seeder* is a client that has a complete copy of the torrent and still offers it for upload. The more seeders there are, the better the chances of getting a higher download speed. A *downloader/leecher* is any peer that does not have the entire file and is downloading the file. Bram chose the term downloader over leech because BitTorrent's tit-for-tat ensures downloaders also upload and thus do not unfairly qualify as leeches. With the adoption of *DHT (Distributed Hash Tables)* the BitTorrent protocol starts to become more that a semi-centralized distribution network around a single resource, it becomes more decentralized and removes the static point of control, the tracker, this is done by relying in DHTs and the use of the *PEX* extension. Enabling the volatile Peer to operate also as a tracker, but even if this addressed the need for static tracker servers, there is still a centralization of the network around the content. Peers don't have any default ability to contact each other outside of that context.



2. CHARACTERISTICS OF BITTORRENT

Permanent DHT tracking:

With the PEX implementation and reliance on the distributed hash table (DHT), the evolution into creating a real P2P overlay network that is completely serverless was the next logical step. The DHT will take information not only from old trackers but also from the PEX implementation, creating something like a distributed Database of shared torrents acting as backup tracker when all other trackers are down or can't deliver enough peers, as well as enabling trackerless torrents. The DHT acts and is added to torrents as a pseudo-tracker if the client has the option enabled and DHT trackers can be enabled and disabled per torrent just like regular trackers. Clients using this permanent DHT tracking are now a fully connected decentralized P2P network, they enter the DHT as a new node, this of course makes it necessary for private trackers (or non-public distributions) to exclude themselves from the participating.

Magnet links:

The **Magnet URI scheme** refers to resources available for download via peer-to-peer networks. Such a link typically identifies a file not by location, but by content more precisely, by the content's cryptographic hash value. Although it could be used for other applications, it is particularly useful in a peer-to-peer context, because it allows resources to be referred to without the need for a continuously available host. Traditionally, .torrent files are downloaded from torrent sites. But several clients also support the Magnet URI scheme. A magnet link can provide not only the torrent hash needed to seek the needed nodes sharing the file in the DHT, but may include a tracker for the file.

The attributes of BitTorrent are Web seeds, PEX , Global and local connections, Tracker URL, Piece hash values, Info hash, File length, Piece length, Bencode where

Bencode is the encoding used by the P2P file sharing system BitTorrent for storing and transmitting loosely structured data. It supports four different types of values: byte strings, integers, lists and dictionaries (associative arrays). Bencoding is most commonly used in torrent files. These metadata files are simply bencoded dictionaries.

Message digest:

A Message Digest is a digitally created hash (fingerprint) created from a plaintext block. All the information of the message is used to construct the Message Digest hash, but the message cannot be recovered from the hash. For this reason, Message Digests are also known as one way hash functions.

SHA-1:

SHA-1 is the most widely used of the existing SHA hash functions, and is employed in several widely used applications and protocols. SHA-1 produces a 160-bit message digest. SHA-1 and SHA-2 are the secure hash algorithms required by law for use in certain U.S. Government applications, including use within other cryptographic algorithms

and protocols, for the protection of sensitive unclassified information.

Disadvantages:

The main disadvantage of the BitTorrent network is that many of the torrents are not accessible to the users participating in the file sharing process. There is no single place to have access to all the torrents in the system. The websites that host or cache the torrent files have some restriction or there is some inefficiency to index all files.

3. PROPOSED SYSTEM

The major disadvantage in the whole BitTorrent system is that there is no access to all of the torrents available and thus there is not much sharing among the peers. Although there are some hamsters/ bots that collect the torrent information from a considerable number of websites which host the torrent files, there is a limitation to this. Another way is the use of torrent caching sites which cache the torrent files on their servers and are accessible only through their hash. There exists many torrent sites that provide torrent cache, but one cannot search through them until they have the hash for the torrent they want. This becomes very much inconvenient for a naive user to search through these sites. One way is to map info hash values of each torrent with the name of the torrent by parsing the torrent file. The hash would be mapped with the torrent names along with a set of URLs and magnet links from where the torrent files can be downloaded and store them in a database from where the user would be able to search for torrents using the name of the torrent. This can be implemented in client software where it will interact with the database on the server or a web based search.

The pre-requisite for such a system would be a strong database capable of handling a large number of records at a given point of time, higher bandwidth internet connection (possibly the bandwidth of a server), and a little bit knowledge of the BitTorrent protocol.

The database can first be populated by mapping the hash value of the torrents and their other key properties and inserting these records into the database. After this step, a search function needs to be implemented that can search the database related to the keywords specified by the user returning the links where the torrent file can be downloaded by the user. If implementing this system as a standalone application software, the software may accept a search string from the user, query the database on the server and return results that are related to the search string specified to the user. If implementing as a web based system, the system can accept a search string from the client and return the results to the client browser.

Through this system, the users are exposed and made accessible to a large number of torrents on the network through which they can share more data and it is accessible to a large number of users in the BitTorrent network.

Advantages:

This proposed system will allow a user to access any torrent uploaded on a website not familiar with the user, making its major advantage of accessing any remote torrent and this will create an efficient system for the required search.

Also, a torrent uploaded on multiple sites will be shown as a single result in our proposed system, unlike other search engines, which provide multiple results for a single torrent.

4. CONCLUSION

In this paper, we have clearly presented the terms and characteristics all of the BitTorrent protocol. The disadvantages of the direct connect protocol are covered in the BitTorrent protocol, still as every coin has two sides, the BitTorrent protocol also must be having its disadvantages. As we can see above in this paper, our proposed system gives access to large number of torrents that might not be accessible from familiar websites, thus allowing an efficient torrent searching for everyone including the naive users too.

5. REFERENCES

[1] Bram Cohen, The BitTorrent Protocol Specification http://www.bittorrent.org/beps/bep_0003.html
[2] http://wiki.theory.org/BitTorrentSpecification#Bencoding
[3] http://en.wikipedia.org/wiki/Glossary_of_BitTorrent_terms
[4] John Hoffman, HTTP Seeding http://www.bittorrent.org/beps/bep_0017.html
[5] J.A. Pouwelse, P. Garbacki, D.H.J. Epema, H.J. Sips, The Bittorrent P2P File-Sharing System: Measurements And Analysis http://www.cs.unibo.it/babaoglu/courses/cas04-05/papers/bittorrent.pdf
[6] http://en.wikipedia.org/wiki/Comparison_of_BitTorrent_sites