Bit Torrent P2P System

Name of Student : Mohit Akhouri

Roll No : 19ucc023

Email : 19ucc023@lnmiit.ac.in

Abstract

P2P (Peer-to-Peer) Network has captured a lot of attention nowadays. A P2P application is a type of software which allows the clients to communicate over a network with each other directly. A P2P system has both client and server and they contribute equally. BitTorrent is a distributed P2P system , and it can be used to broadcast media and can be used in file distribution. Bit Torrent relies on the principle of Tit-for-Tat (symmetric transferring model). There are various mechanisms and algorithms in BitTorrent System which ensures that every downloader is equally treated. A BitTorrent base its operation around three things – A torrent file, A centralized tracker and an associated swarm of peers. The centralized tracker basically provides the address list of live peers to different entities. Using the BitTorrent Protocol, large servers can be replaced by home computers while efficiently distributing files. The major issues in these types of P2P systems are of scalability and reliability which we will discuss.

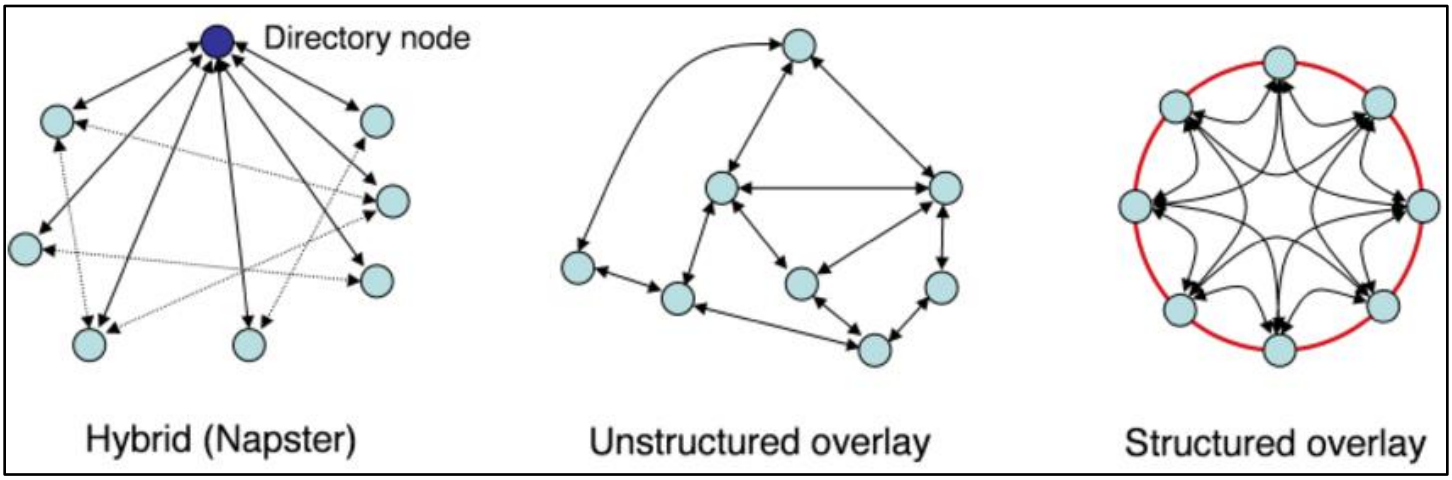
This term paper will focus on the BitTorrent as a peer-to-peer system and will explain the architecture and concepts behind it. Later in the term paper, we will also analyze the security issues while working with BitTorrent Systems.

Background / State-of-the-art

Before understanding about BitTorrent, let us look at some of information about **Peer-to-Peer** networking in Distributed Systems. If we try to define Peer-to-Peer networking in simple terms, it can be defined as : " It is a communication model in which each participating party has the same capabilities and any of the party can initiate a communication session." P2P networks is a distributed system and it partitions tasks/workloads between the participating entities, which are called peers. There is minimal or no reliance on the servers in a P2P network. Unlike the Client-Server architecture, peer-to-peer architecture encompasses decentralized network of peers as nodes which can play the role of client/server with their corresponding duties and capabilities. The peers further could be divided into super peers and edge peers. For Example : Super peers can come up with more than they can consume mass resources and Edge peers can only consume and do not donate to the operations performed by the network.

There are different types of Peer-to-Peer systems which are as follows :

1. Structured P2P Systems
2. Unstructured P2P Systems
3. Hybrid P2P Systems



In **Unstructured P2P** , the overlay network organizes the peers in a random graph in flat or hierarchical manner. In **Structured P2P**, the overlay network are based on a data structure called the Distributed Hash Table (DHT's). In this type of P2P systems, the network assigns keys to data items and organizes the peers in a graph , mapping each data item to a peer. Data items are also assigned some identifiers from a large identifier space. **Hybrid P2P** is basically a combination of the peer-to-peer architecture and client-server model. Hybrid P2P is beneficial when we need a central server with P2P capabilities.

Introduction to BitTorrent : BitTorrent is one of the most popular application that support peer-to-peer architecture. The main idea behind BitTorrent was to store files at a safe location by splitting it into pieces at many locations. BitTorrent is free and open source software. BitTorrent employs the tit-for-tat policy, which means it slows down the download rate for the peer which is only downloading and not uploading anything. BitTorrent accounts for about 35% of the Internet traffic.

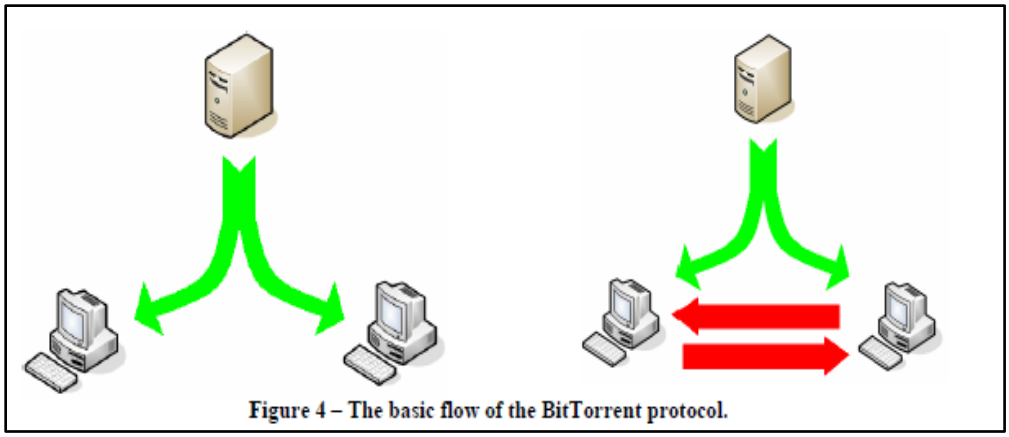


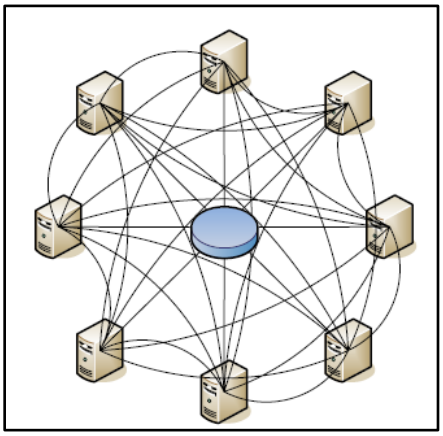
Figure 4 – The basic flow of the BitTorrent protocol.

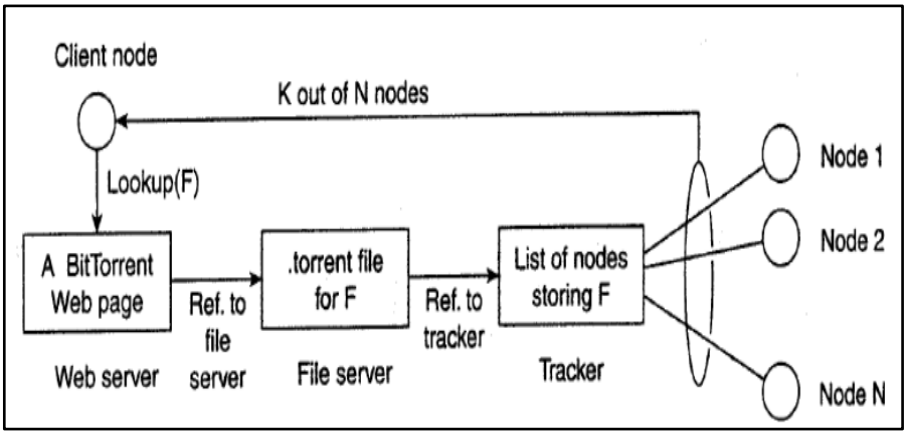
The figure on the left shows a client-server architecture for downloading. The figure on the right shows the BitTorrent approach. The file is split and one part is sent to each peer and let the peers download the part which they are missing from one other.

Basic Terminology in P2P systems : Some of the most common terms that we come across in P2P systems are defined as follows :

1. *Peer Set* : The peer set consists of a list of peers that participate in the network.
2. *Piece and Block* : Data in P2P systems is split into smaller pieces which are distributed among the peers and further each piece split into blocks, which are transmission units in the network.
3. *Metainfo file* : It is a file which consists of all the necessary details of the operating protocol.
4. *Torrent* : It is a type of file which facilitates peer-to-peer sharing.
5. *Current Swarm* : The piece which is held by the lowest number of peers.
6. *Seed* : A seed is a BitTorrent user who has 100% of a file and is sharing the file with other users for downloading.
7. *Leech* : It is a BitTorrent user who downloads the files shared by the seeds and does not seed back to other users.

Architecture of BitTorrent : The BitTorrent architecture mainly consists of the following entities : a static metainfo file (which is also knows as torrent file), a tracker, an original downloader (seed) and the end user downloaded (leech). BitTorrent in its original form matches the Hybrid P2P architecture.





The basic idea behind the working of BitTorrent is as follows. When the end user looks for a file, he downloads the chunks of file from other users until the downloaded chunks can be assembled together and yielding the complete file. To download a file , the user needs to access the global directory which consists of the references to metainfo files also knows as torrent files. Torrent files contains links to a tracker which keeps the count of *active* nodes that have the chunks of requested file. When the nodes have been identified from where the chunks of requested file can be downloaded, the requesting node becomes active and an algorithm forces it to help other. The enforcement ensures that : if P and Q are two nodes and Q sees that P is

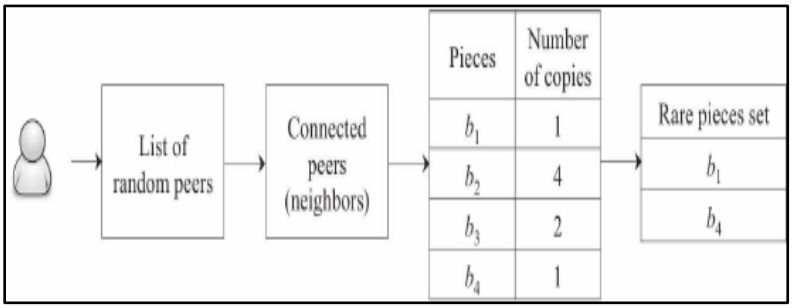
downloading more than it is uploading, then it decreases the rate of download for P. To download a file , we need a BitTorrent client which is a free application that administrates the download procedure. The main working of BitTorrent application lies behind some algorithms. Choosing peers for downloading files is a two-sided task. The two steps that are employed in choosing peers are :

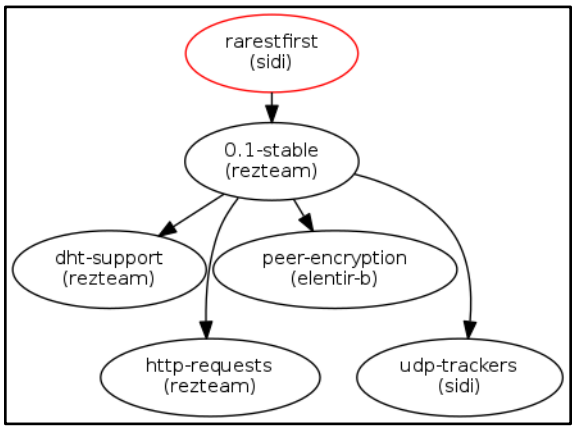
1. Find the **best sequence** of downloading the **required chunks**. This can be accomplished with the help of **Piece selection algorithms**. Some of the Piece selection algorithms are : Rarest first, Strict Policy, Random first piece and Endgame mode. We will look at the rarest first and random first piece algorithm later in this paper.
2. Next comes the **resource allocation**, when some peers do not allow other peers to download from them. The algorithms behind them are known as **choking algorithms**.

The details of the Piece selection algorithms are choking algorithms are discussed below.

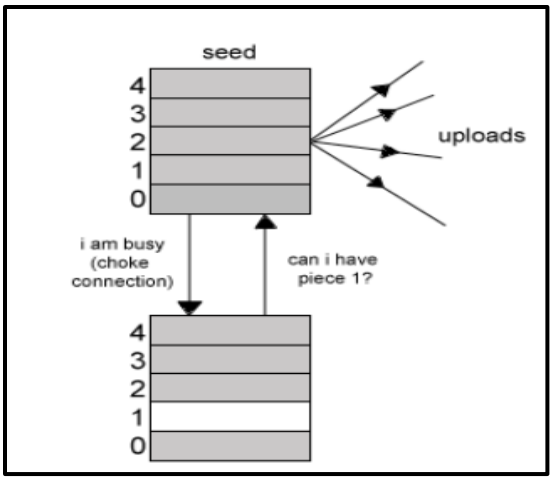
Piece Selection Algorithm : Piece selection algorithms ensures that there is a correct selection of pieces so that all peers may not end up having downloaded the same pieces. The two policies which we can follow for Piece Selection are as follows :

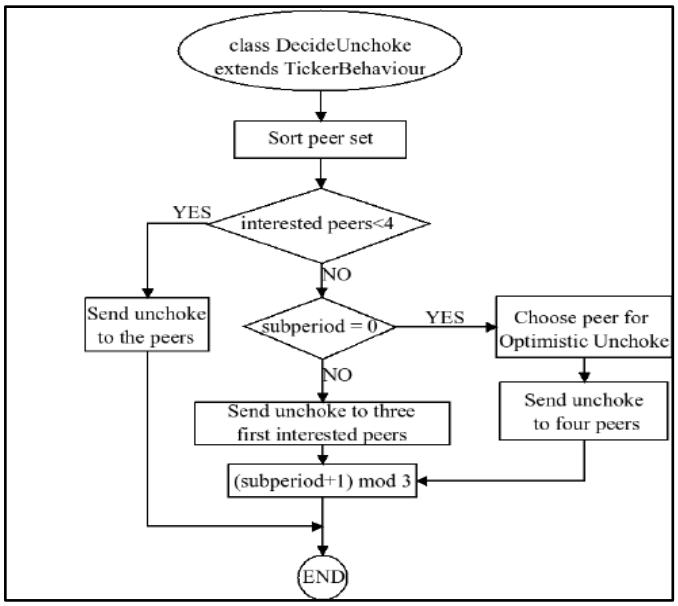
1. *Random First Piece Selection* : In this selection policy, random selection of the piece of file is chosen and is downloaded as soon as possible.
2. *Rarest Piece First* : In this policy, the algorithm determines the pieces that are not found or are rare among the peers and they are downloaded first. This algorithm ensures that common pieces are left towards the end to be downloaded.





Resource Allocation : Resource allocation is needed in BitTorrent systems because there is not centralized resource allocation. Every peer is responsible for maximizing it's download rate. The choking algorithm is employed so as to avoid free riders, which are those peers who only download and do not upload anything. Now to determine which peers to choke and which to unchoke, a peer unchokes a fixed number of peers, the default is four. BitTorrent allows an additional unchoked peer, where the download rate criterion is not used. This method is known as optimistic unchoking.





The above two diagrams illustrate the choking algorithm employed in BitTorrent P2P systems. Next we will discuss the applications of BitTorrent P2P system. We will also look at some areas of usage of BitTorrent P2P systems.

Application of BitTorrent P2P Systems : Some of the applications of BitTorrent P2P systems are as follows :

- BitTorrent is employed in µTorrent, rTorrent and deluge. These applications uses the file sharing protocols and P2P architecture.
- Since in P2P systems, there are lot of active peers new peers can attach themselves to other peers easily.
- BitTorrent P2P systems can be used for file sharing, instant electronic messaging, auditory communication, collaboration and high performance computing.
- The workload is reduced from individual peer and is distributed among various peers. If a file is broken down, the several chunks of file from different file can be transferred to a peer at the same time.

Areas of Usage : Some of the areas of usage of BitTorrent protocol are as follows :

- We can download Linux Distributions using BitTorrent which is a faster download than the regular FTP or HTTP can provide. We can also download the Internet Browser Opera using BitTorrent.
- With increased download speed, BitTorrent eases the pressure on individual servers whenever a new version is downloaded.
- BitTorrent also has a business usage. ISO-images, operating systems, patches and large software can be downloaded using BitTorrent at a high speed.
- BitTorrent can also be used withing an organization to distribute the applications and updates more rapidly.

Case Evaluation

Although BitTorrent has a lot of usages, but it becomes a legal and security risk when it is used for downloading large files. Torrenting can prove to be a high risk business. We may come across viruses and spyware and there can be legal issues also due to copyright infringement. Some of the **Security Issues** regarding BitTorrent are discussed below:

- ***Malware risks*** – Torrent files which we have just downloaded may contain a virus. Such a virus can easily cripple down the computer system of the user. One of the solutions to tackle this problem is always scan a torrent file after downloading it. We should use a trusted anti-virus program for asses and check if the file is infected with malware content. Sometimes, trackers also filter and remove torrent files which are reported to contain malicious content.
- ***Data Safety*** **–** Another fear using BitTorrent system is the unauthorized access to organization's critical business information. The critical information may include employee records, organizational policies, information about the company's stakeholders and compensations. One solution for this can be to make a separate folder for torrent files.
- ***Vulnerability*** **–** Whenever we go online or download torrents, we become vulnerable to hack attacks. When a user downloads, uploads or share a torrent file, the information about the user's activities can be collected by the hackers. Virtual Private Networks (VPN's) can serve as a solution to this problem. Using VPN, we can do anonymous torrenting.
- ***Legal Troubles*** **–** Some torrent files that are shared among peers can contain some copyrighted material. Downloading such files can land the user in legal troubles. The ISPs also monitor the torrent user's activities online and if they find any trace of illegal torrent file downloads or any copyrighted material, the users are liable to face legal consequences. Legal action may lead to issue of DMCA-like notices, hefty fines and suspension of ISP license. One should check for a legal torrent file before downloading.

There are some other issues in BitTorrent which require further improvement. BitTorrent in itself is **not suitable for multimedia streaming** since it does not the meet the real-time needs of streaming applications. The first problem is that peers do not download chunks in sequence which leaves then unusable until the download is complete. The tit-for-tat policy forces too many peers to wait for a long time before joining the swarm of downloaders.

Another refinement is needed for the **large traffic** the BitTorrent faces. The large volume of BitTorrent traffic can easily have negative impacts on real-time traffic such as VoiceOverIP or VoIP. The files which are transferred in BitTorrent are large in size ranging to hundreds of megabytes and it can prove to be major problem for a LAN with many users. Secondly, the **centralized tracker** can make the BitTorrent network less fault tolerant.

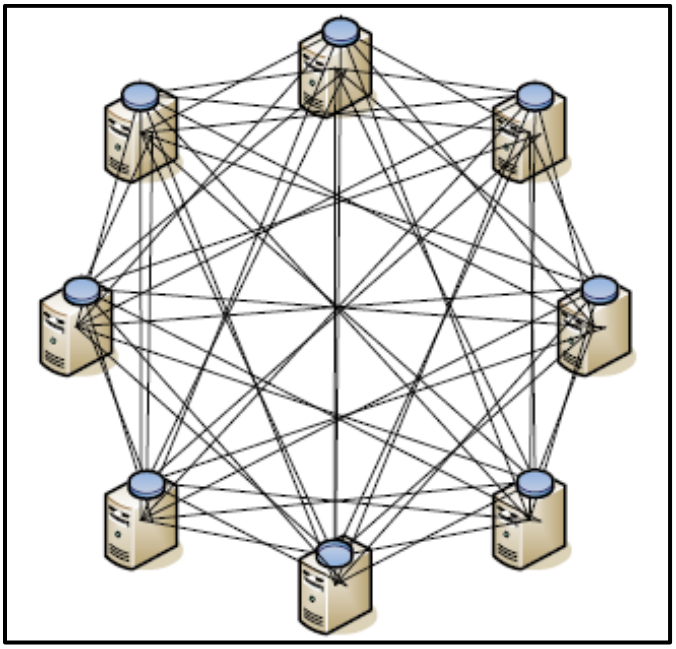
Scope and Future Work : The number of followers of BitTorrent are very large. It can be predicted that BitTorrent represents a technology which can change the broadcast media and file distribution landscape. The future work regarding BitTorrent has already started. There are applications which combine BitTorrent with RSS (Really Simple Syndication). We can construct an overlay tree so as to minimize the end-to-end delay and maximize the utilization of bandwidth of peers. The traditional overlay tree scheme can be made resilient by the duplication of chunks along a small number of randomly chosen additional overlay links. Multi-tree approaches can also be proposed that make use of multiple distribution trees and can assume the presence of uninterested nodes to forward traffic. In the next section, we can look at some of the Proposed solutions to the problems discussed in this section.

Proposed Solutions

Some of the important improvements that need to be made to tackle the problems discussed in the previous section are as follows :

1. ***Bulk Traffic Marking*** – To tackle the problems of heavy traffic, the traffic can be marked as bulk traffic and then we can use any standard traffic shaping tool to manage the network traffic. This makes BitTorrent more business friendly.

2. ***Decentralized tracker*** – In this type, the tracker is distributed such that every client in the network can work as a lightweight tracker. The decentralized concept can make use of a data structure called as Distributed Hash Table or DHT. Decentralized trackers makes it possible to share files with minimal resources but the problem of reliability still exists. The BitTorrent with decentralized tracker structure is shown in the figure below.



3. ***Chunk Selection Policy*** – To maintain a window of chunks of video at application level is one of the common approaches in multimedia streaming. It smooths out the playback jitters. To implement this technique, we can use a sliding window approach. The sliding window contains the next **w** chunks which can be consumed by the client. We can prevent peers to request chunks located outside their current window. This additional constraint on the chunk selection policy can decrease the efficiency of BitTorrent Protocol but can also improve the quality of delivered multimedia stream.

4. ***Neighbor Selection Policy*** : In BitTorrent system, the peers select the other peers by observing their behaviors. The behavior depends on the observed download rates and this prefers the faster peers so as to speed up the process of chunk propagation. If we want to use BitTorrent for real-time multimedia streaming, the **bootstrapping process** needs to be speeded up. We can use a randomized policy which states that at the beginning of every playback, each peer can select neighbors at random for the *randomized choking interval.* Finally, when the peers have chunks to exchange tit-for-tat policy can be used to tackle the free riding effect. Since more peers are able to actively participate during each playback interval, the efficiency of the system improves.

Conclusion

The BitTorrent has now become one of the most popular protocol for peer-to-peer applications. Main reasons for its popularity is because it is easy to use and understand and the download and upload rates are very high. It can be used as a powerful file sharing tool. Replacing the central tracker with a distributed tracker makes BitTorrent a pure peer-to-peer protocol. Another characteristic of BitTorrent is that it is highly self-configuring, that is peers find other peers through the help of algorithms and central/distributed tracker and do not require any human intervention. BitTorrent can leverage the costs of movie and show distribution and can make broadcasting possible for every Internet User. Some of the Advantages and Limitations of BitTorrent protocol can be summarized as follows :

Advantages : Some of the advantages can be as follows :

- BitTorrent is economical and uses the offered resources as central server.
- It is cost effective since they do not need server infrastructure and bandwidth.
- If the network is large, there are very less chances of failure of the network.
- Maintenance cost is very less for BitTorrent network.
- Efficiency is also high since every entity of the network is a content provider. There is no dependency on a single entity for providing service.
- The work is evenly distributed among the peers.

- Adaptability is there in BitTorrent network to survive attacks.
- There is a flexibility to alter the already existing peers. We can add/remove peers from the network.
- Self scalability is there in these networks.

Disadvantages : Along with the advantages discussed above, there are some disadvantages as well which can be as follows :

- There can be exchange of only a part of the whole application, if no seeder exists.
- Peers are dependent on one another for bandwidth.
- BitTorrent network cannot be used for private sharing, since it requires the involvement of different users and sharing among them.
- As discussed in the security issues, there is a issue of copyright infringement. Some of the contents in BitTorrent network has copyright material which can cause legal issues.
- If a single peer is affected by malicious content or virus, it can damage different receiving peers.
- There can be a stress on ISP, since BitTorrent networks are designed for asymmetrical information measure usage for downstream traffic rather than upstream traffic.
- The BitTorrent network relies on the bandwidth, storage and computation resources volunteered by the users.
- There can be access permissions if some confidential files are shared with high quantity of peers.

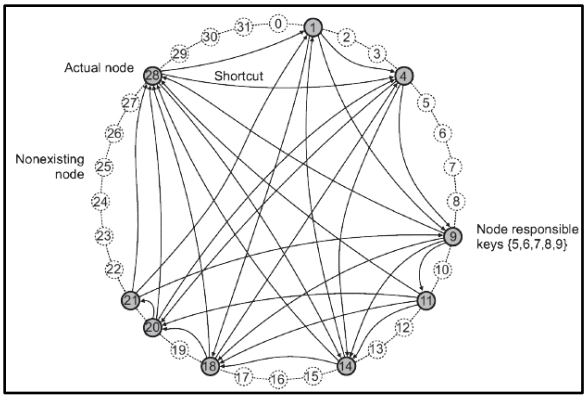
Implementation

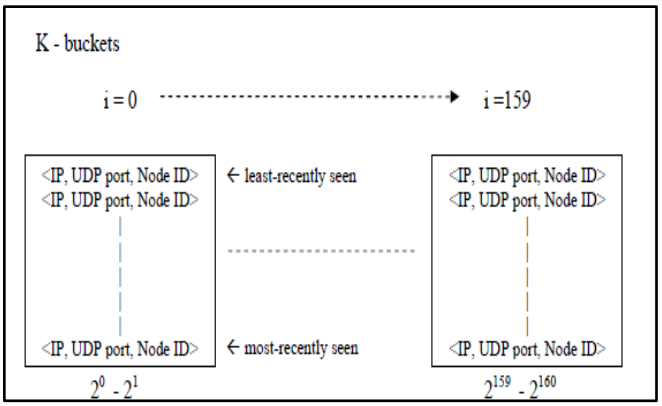
We can make some improvements to the BitTorrent network by Bulk traffic marking, which can reduce the heavy traffic. We can use any standard traffic shaping tool for this. Another improvement we can make is replacing centralized tracker with decentralized tracker. The decentralized tracker can be implemented with the help of Distributed hash tables or DHTs.

Distributed Hash Tables : The DHT can perform like a hash table. In DHT, key and their corresponding value can be stored and later lookup can be made regarding a particular key. The storage of key-value pairs and lookup operations are distributed among the participating peers of the network. The nodes can freely join and leave the network. The DHT protocol has the following characteristics : Decentralization, Scalability and Fault tolerance. The DHT structure consists of two parts which are keyspace partitioning and overlay network.

Keyspace partitioning : Every node in the BitTorrent network is assigned a key also known as identifier. Concept of consistent hashing exists in DHT. In consistent hashing, the mapping of keys to buckets do not change significantly. When the ownership of keys is changed, there is a

bandwidth intensive movement of objects from one node to another. An example of Keyspace partitioning can be shown in the figure below.





Kademlia : It is a commonly used overlay network protocol used in decentralized peer-to-peer application. It has a wide range of features compared to previous peer-to-peer applications. The number of configuration messages is minimized between nodes for the purpose of node detection. The nodes in the BitTorrent network using Kademlia communicate using UDP protocol.

References

1. Maarten van Steen, Andrew S. Tanenbaum, Distributed Systems, Third edition, 2017

2. Jahn Arne Johnsen, Lars Erik Karlsen, Sebjon Saether Birkeland, Peer-to-Peer networking with BitTorrent, Department of Telematics, December 2005.

3. Purvi Shah, Jehan-Francois Paris, Peer-to-Peer Multimedia Streaming using BitTorrent, University of Houston

4. Sandhya Shanvbag, Venkat Chavan, Peer to Peer Networks: A Bit Torrent Case Study, Department of Information Science and Engineering, Bangalore, June 2021

5. Ken Kyoungwoo Nam, Analysis of BitTorrent Protocol and Its Effect on the Network, Spring 2011

6. Security-info BitTorrent, https://www.pandasecurity.com/en/security-info/bittorrent/

7. 4 risks of using torrents, https://www.creative.onl/startupsgeek/4-risks-of-using-torrents/, November 25, 2020