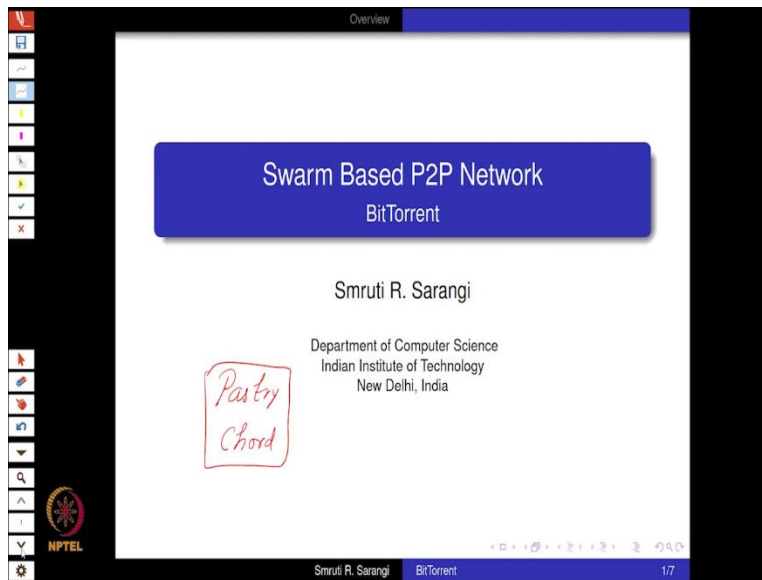**Advanced Distributed systems**
Professor Smruti R. Sarangi
Department of Computer Science and Engineering
Indian Institute of Technology Delhi
Lecture 06
**BitTorrent: Large scale file sharing**

(Refer Slide Time: 00:17)

Welcome to the ultra-short lecture on BitTorrent. So, BitTorrent is one of the most popular file sharing programs peer to peer, network-based file sharing programs as of 2021. And this is based on the familiar technology of DHTs. So, this is an ultra-short lecture, but before you go forward, I would request all of you to take a look at the videos for Pastry and Chord, that are a part of this lecture series, because without understanding Pastry and Chord, this lecture will not be comprehensible.

So, first, take a look at that. And we will only outline few of the basic points, salient features of BitTorrent, the rest will be fairly clear to somebody who understands distributed hash tables.
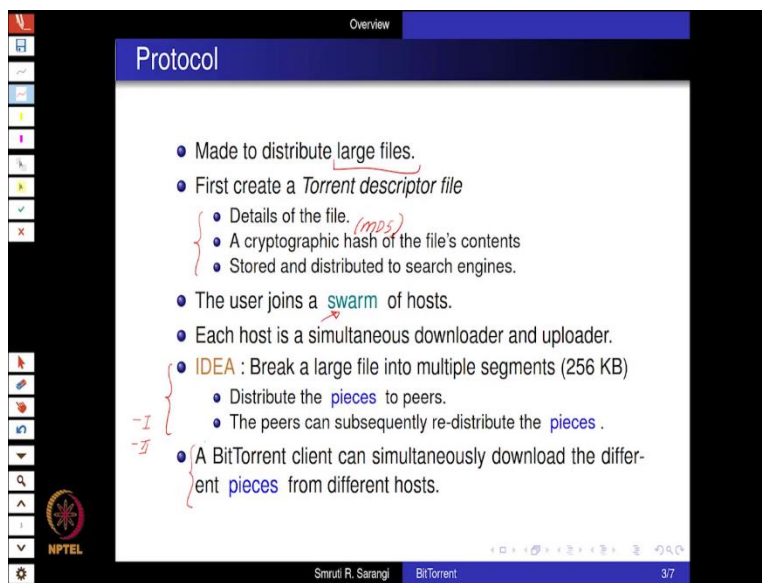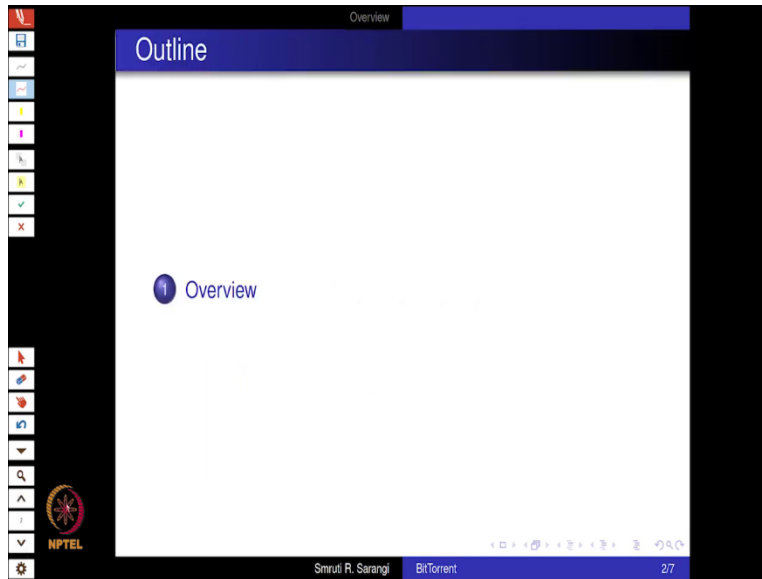
So, overview, so as compared to Napster and Nutella that were typically for smaller files like mp3 files and music files, BitTorrent was made to serve large files, or large video files, so it was the main aim was to distribute large video files. So, because of that, it is necessary to kind of re architect our system. So, the user first, the user who is sharing the file first creates what is called a torrent descriptor file.

The torrent descriptor file has the details of the file, a cryptographic hash of the files contents. So, the cryptographic hash is required for integrity, because since we are talking of a large file, and

we will discuss how it is actually served, it is possible that some bytes may develop a fault. So, because of that a cryptographic hash is required. So, typically, the MD5 hash is used for this purpose. And then it is stored and distributed via search engines or via a peer to peer system.

So, the user joins a swarm of hosts, it can simultaneously be downloaded and uploaded. So, we have been seeing the same format. In other P2P systems as well, we are seeing the same, we have seen the same in Napster, same in Nutella, that the user actually shares a shared directory where you have songs and videos and so on. So, since we are talking about large files, such as videos, we break a large file into multiple small segments, so each segment is 256 KB.

And these are distributed to peers. So, these pieces of files are distributed to peers. So, this allows the client, the BitTorrent client, to actually download all of these segments in parallel. So, this increases the bandwidth and also reduces the time needed to get a file. And furthermore, it increases the robustness of the system. So, the peers can themselves redistribute the pieces. So, this will further add to the robustness.

And pretty much for every file, we will number the segments, 1, 2, 3, 4, and so on. And we know how many segments that are. So, these segments will then come from different parts of the network. So, the BitTorrent client, which is a piece of software that every user needs to install, can simultaneously download the different pieces from different hosts.

(Refer Slide Time: 03:42)

So, the key elements in BitTorrent are like this one is a torrent file, which contains the metadata, metadata means a description of the file that will be used for searching, and the hash. And then we have a specialized entity called a tracker, which is a server. So, this used to be pretty popular in the early days of Bitcoin. So, the tracker was coordinating the entire process of downloading a file.

This means that the approach would be to connect to the tracker, so the client would connect to the tracker server. This would have a list of peers that contain the different segments. And then the client would connect to the peers to get the different pieces. But now the tracker has gone away mainly because of legal issues, you do not want to have one server which has the list of the entire network.

So, instead of the tracker, this has been replaced by a DHT. And the DHT will help you locate all the peers that contain a given piece using our same DHT mechanism that we are studied in Pastry and Chord. And the DHT that is used is called the mainline DHT which basically uses the Kademlia protocol. So, recall that in the last few slides of the Chord lecture, we did discuss that Kademlia protocol to a certain extent.

(Refer Slide Time: 05:02)

So, downloading and sharing files, well users need to use regular search mechanisms to find torrents of interest. Similarly, if a server has a new file, it will host it and distribute the torrent file. So, it is known as a seeder. So, once the client finds a torrent file, it will connect to the tracker, or

it will use the mainline DHT and will download the pieces in a random order. So, there is no fixed order.

So, you can download piece three, piece one, piece two, piece four in any order. So, there can be different strategies. So, we can prioritize traffic for those nodes that have sent a lot of data on the network. See, if let us say that I have been a very active uploader. In a sense, I have been very actively supplying my files, I should get some priority while downloading. And also, tit for tat relationships in the sense of I gave you something then you will also give me something back with a high priority.

And furthermore, I can reserve some bandwidth for myself and have some bandwidth for others. So, the main problem actually that happened with Bitcoin is that again, we go back to college students, so what they were doing is that they were sharing a directory and the directory used to have these files and their associated torrent files.

So, all day others were downloading unbeknownst to the sharer, so that was eating up a large part of their bandwidth. And when they wanted to download, they did not have enough bandwidth. So, some of that, so modern clients are configurable, so some bandwidth can be reserved for oneself, and some for others.

(Refer Slide Time: 06:47)

Security and privacy, so as such BitTorrent does not provide an anonymity or security. And furthermore, the onus is on the site that indexes the torrents like a tracker site. Even without that everybody involved in the hosting and propagation of copyrighted or illegal material, in a sense is legally culpable. So, of course, to what extent it is enforced depends on the laws of the specific country.

But there are two broad approaches. Either we use a tracker server that provides a directory or we use a DHT. And then the problem with the DHT is it will require multiple hops, but again, the legal liability is much lower. And furthermore, there is more robustness as well as it is easy to locate. And given the fact that will take proportionally much longer time to download the entire segment.

Locating a note that has the torrent file will not take that much time. Plus, these are not strictly real time tasks. So, we do not really worry, need to worry about the latency to that extent. So, BitTorrent as of today is banned in a lot of places, particularly university campuses, regardless of whatever you are using tracker or DHT. So, that needs to be understood.

(Refer Slide Time: 08:07)

In spite of that BitTorrent is extremely popular. So, the mainline DHT, which BitTorrent uses is the largest DHT in the world, so it does have somewhere between 10 million to 25 million

connected computers. So, BitTorrent is clearly the largest file sharing system in the world at the moment.

And all the current versions of the BitTorrent clients are compatible with mainline DHT but they can connect to trackers as well. Furthermore, BitTorrent is expanding, or rather, I would say has expanded and it uses other kinds of protocols. For example, it uses a gossip-based protocol. So, basically to synchronize BitTorrent directories, to implement BitTorrent directories among the peer nodes. So, this protocol is called Tribler.

So, this is again, a gossip-based thing where I just maintain a directory of file names and servers, and we periodically exchange an update. So, go back to the lecture on epidemic and gossip-based algorithms. So, we use anti entropy to regularly extend the list of torrents. And furthermore, since there are lots and lots of torrents, the BitTorrent software also gradually learns about the user's preferences and filters, the torrents and essentially stores those torrents that are more aligned to the user's viewing preferences.

So, this in a nutshell, was BitTorrent, we did not discuss much about the Kademlia protocol or the mainline DHT. But my feeling was that whatever we discussed was the end of code is enough to give an introduction to Kademlia. And the protocol of course can be read up on the web. But the main idea with BitTorrent should be clear that it is clearly the largest DHT in the sense that runs in the world and it uses other methods also that include gossip-based algorithms and trackers.

So, the BitTorrent Wikipedia article can give a quick introduction. If you want to know more about BitTorrent, you can always read this paper by Izal, Mikel, et al and it talks about 5 months in a torrent's lifetime, so it will tell you everything about it. So, this lecture pretty much finishes our discussion on DHTs, we have discussed quite a few, we have discussed Pastry, we have discussed Chord, we have discussed, tapestry, Kademlia, one slide each and now we have discussed a system made on a DHT, the mainline DHT, the BitTorrent system.

So, subsequently we will move to the second half of the course. So, the first part was essentially DHTs and epidemic gossip-based algorithms and so on. So, the second half of the course will basically look at distributed algorithms. And that is important because once we have DHTs are only one kind of a distributed algorithm, but there are many more types and all of them are required.

And finally, we will use the results of parts 1 and 2 to create actual systems. We did see one actual system, BitTorrent is an actual system, but we will create bigger systems that use the results taught in parts 1 and 2 of this course.