

# Coordination Among Clients Improves Robustness in Bittorrent

Laxman Vembar  
Dr.Ambedkar Institute of Technology  
Bangalore  
Email: laxmanvembar@gmail.com

**Abstract--This paper proposes a new mechanism of coordination among bittorrent users allowing people to coordinate their download with their friends or other members of an online community. Doing so improves the robustness in the system giving users a greater chance of successfully completing their download without hampering their potential download speeds.**

## 1. Introduction

Bittorrent is a p2p file sharing protocol which has become increasingly popular to transfer large files over the internet. It has been designed to be an efficient content distribution mechanism rather than an anonymous p2p file sharing system. This is especially suggested by the fact that bittorrent does not attempt to provide any mechanism for content discovery. Bittorrent however still considers the downloaders as an anonymous group and does not provide means of developing trust relationships among the peers involved in a download process. Providing such a mechanism would improve the robustness in bittorrent by allowing groups of users to coordinate their download(with their friends or other members of an online community) so as to ensure that, among themselves, they have the complete source of the file in the minimum amount of time.

## 2. Why use a coordination protocol?

All bittorrent users, at some point, encounter the problem of dry torrents. This is essentially a situation wherein the original source of the file has now disappeared and some parts of the file are now not available in the network. This would result in everyone downloading close to completion but no one being able to complete the download because they cannot find sources for the missing parts. At this stage the downloaders have no option but to either terminate the incomplete download or hope that that someone with the missing parts will join the download process.

This problem has been in some ways overcome currently by the use of private trackers. The tracker is essentially the only central link in the bittorrent process which maintains a list of peers who are trying to download the same file. By making this private, every

user now needs to register with the tracker before he can use the tracker functionality. This system allows the users to contact each other by any means provided either directly by the site or secondary means such as email etc. Thus when we have the problem of a torrent drying up, the users currently downloading the file can request a user who has already completed downloading to join the download group and make the missing pieces available.

The drawback of using this system however comes from the fact that every user who wishes to use the private tracker must be registered to the tracker in the first place. Thus the user who acts as the original source for a file is limited to the group of users who have registered with the tracker. This has resulted in the development of several bittorrent communities each of which uses its own private tracker.

The coordination protocol primarily attempts to address this problem and tries to create a reliable source for the file from an unreliable one rather than starting with a reliable and accountable source in the first place(as in the case of private trackers).

## 3. Creating a “Virtual Seeder”

A “seeder” in bittorrent terminology is any person who has the complete file. That means to say that he has all the pieces of the file. Users in a bittorrent swarm however, will be able to complete the download even if there are no seeders, as long as all the pieces of the file are available. Whether the source of these pieces is a single peer or a group of peers is immaterial. What the coordination protocol attempts to do is to try to obtain all the pieces of the file and thus create a “virtual seeder”. A “virtual seeder” is thus a group of peers associated with a bittorrent task, who among themselves, have all the pieces required to reconstruct the complete file. In essence we are trying to create a group of users who have pieces that are mutually exclusive and collectively exhaustive.

## 4. Main Considerations

As stated before, the primary goal of the coordination process is to ensure that all the parts of the file are available within the users involved in the coordination process in the minimum amount of time. The Coordination Tracker achieves this by advising the

clients on which parts of the file to set higher priority to thereby ensuring that the users involved in a coordination process always download mutually exclusive parts of the file. There are however a few major considerations in implementing this protocol.

- The process must be simple
- The process should not reduce the efficiency of the bittorrent protocol
- The process should be beneficial not only to the users involved in the coordination task, but to the other peers as well.

The main aim in trying to keep the process simple is to reduce bandwidth requirements. All information connected to the coordination process is ultimately overhead and hence this data must be minimal allowing maximum bandwidth for the actual data transfer.

Also, the process should not compromise the download speeds available to clients involved in the process. This means to say that clients should not end up in a situation where they get consistently lower download rates because they are connected to the coordination process. Lastly the process must try to ensure that the advantages of adopting the coordination process, is felt by clients outside the coordination group also.

## 5. The coordination process

The coordination mechanism allows users to create a new task and subsequently join or quit the task. Based on information about which parts of the file are currently available within the group, the coordination tracker advises clients on which pieces of the file to give higher priority to.

### 5.1 Initiating a new task

When a user registered with a coordination site wants to initiate a coordinate task, he is required to upload the torrent file and specify the maximum number of users associated with the task. This value cannot be changed at a later time. The coordination tracker proceeds to read the torrent file and obtains information about the number of pieces in the file. Next the pieces in the file are divided by the maximum number of users (as specified by the user initiating the task) and they form the piece ranges in the coordination task. The last range may be bigger than the other ranges. (For example, if a file has 100 pieces and the task allows a maximum of 3 users, then the first and the second ranges will have 33 pieces each and the last range will have 34 pieces.) The user initiating the task is added to the coordination process and the torrent file is also stored locally so that other users wishing to join the task can obtain it easily.

### 5.2 Joining a coordination task

When a user attempts to join a coordination task, the coordination tracker first checks to see if he has been authenticated. If he is a registered member of the site,

he is added to the list of users involved in the coordination task. The coordination tracker can optionally also require the task initiator to certify all new users joining a task.

### 5.3 Quitting a coordination task

A user can quit a coordination task he is involved in at any time. The coordination tracker first ensures that he is authenticated and involved in the coordination task before removing him from the list of active users. Also, the tracker checks the piece ranges to see if any range has been marked as being available with this user. If true, the corresponding piece ranges are marked as unavailable.

### 5.4 Server Behaviour

The coordination tracker essentially maintains information about the state of piece ranges in a coordination task and advises the clients on which range of pieces to give higher priority to. Piece ranges in a coordination task are in one of the following three states at any time.

- *Completed*: Indicating that this range of pieces is available with at least one client in the coordinated group
- *Downloading*: Indicating that one or more users in the group are currently trying to download all pieces in this range.
- *Unavailable*: Indicating that no user in the group is currently trying to download pieces in this range.

Additionally if a piece range is in downloading state, information about the number of users trying to download the piece is also stored. In the case of a completed piece range, the id of the user who verified having all the pieces in the range is stored. When a user in the group requests for a piece range, the server first tries to find an “unavailable” range from the list of ranges. If no such ranges are present, the ranges in the downloading state are checked and the one with the least number of users downloading is assigned. If all the pieces are available, then the server responds indicating that the coordination process is complete and the client can use the regular bittorrent priority mechanism to proceed downloading pieces.

### 5.5 Client Behaviour

The communication between the client and the tracker involves the client first requesting for a piece range to download. On getting a response from the server, the client sends an accept message to the server indicating that it is now actively trying to download all pieces in the given range. On completing the task, it sends a completed range message to the server indicating that it has all pieces in the range assigned to it. If the client finds that the coordination task is complete, it may still

continue to query the server at regular intervals in case a user quits the task making a piece range unavailable within the group.

### 5.6 Setting Priority

In setting higher priority to the pieces in a selected range, the client needs to ensure that it does not compromise the download speeds available to the client. The Bittorrent protocol at most times follows a rarest first strategy of downloading pieces. That is, among the clients connected to the peer, the piece that is available with the least number of peers is given highest priority. A coordination client also uses the same strategy but at two levels. All the pieces within the coordination range have higher priority than others. However among them the rarest first principle applies. Another problem arises when the peer cannot find sources for the pieces in the range it is interested in. In such a case, the client requests pieces outside the range also so the download speeds are not compromised. There is also the problem of not having enough users involved in the coordination task. That occurs when say a task with a “max users” of 20 has only two users. In such a situation, looking for pieces in only two of the piece ranges (ie  $1/10^{\text{th}}$  of the file) may not be optimal as rarer pieces in other ranges may be lost permanently. To overcome this problem the priority is weighted between the values 0 and 1. A value of 1 indicates that 100% of the pieces requested will be in the specified range and one of say .50 indicates that 50% of the pieces requested will be within the range. It is important to note that even with a weight of 1, if the client finds that a peer does not have pieces in the range it is interested in, it will still request for pieces outside the range to keep the download speeds optimal. Here is an example of how the system works. Consider a file with 100 pieces. Let the client be interested in piece range 1 which includes pieces 0 to 9. Let the weight assigned to the coordination task be .50 (i.e. 50% of the pieces requested from each peer will be within this range.) Now consider two peers, A and B. Let us presume that peer A has all pieces from 0-9 and peer B has only two pieces, 0 and 5. Let us also presume that the client wants to request for 10 pieces from each of these peers. Now when the client connects to peer A, after the initial handshake, it finds that the peer has all pieces from 0-9. So now it requests 5 pieces in the range 0-9 and 5 pieces in the range 10-99 both using the rarest first strategy. (ie the rarest piece in the selected range is the first piece to be requested and the rarest piece outside the range is the 6<sup>th</sup> piece to be requested.) Now when the client connects to peer B, it finds that the peer only has two pieces in the selected range. It proceeds to request the two pieces in this range and then continues to request pieces outside the range if they are not

available locally. Thus the potential download speed of the client is never compromised by being involved in the coordination process.

Finally, once the coordination process is complete the client still has to obtain all the pieces using the regular bittorrent protocol since no special mechanism of distributing the pieces among the coordination group is provided. This prevents the users in the group from disconnecting from the swarm and thus provides a reliable source for all the pieces of the file to peers outside the coordination group also.

### 6. Conclusion

The most important criterion in implementing any sort of coordination mechanism is to ensure that the use of the mechanism does not degrade the performance of the client. By ensuring that the client always functions at its full capacity, the only drawback in the process is the overhead caused in transferring messages related to the coordination process. However the bandwidth utilized in transmitting these messages can be considered insignificant when compared to the actual data transfer required to download the file. In preliminary tests, we have found that even in a coordinated group with only three users, the time it takes to acquire all the pieces of the file can be as low as half the time it takes the fastest user to complete the file download. The time can be further reduced if more people are involved in the process and more sources of the file are available. In any case, since all clients always try to download at full capacity, the performance is never hampered. On the other hand, the performance of the swarm in general is enhanced somewhat because of the fact that we have more sources for all parts of the file rather than have a few pieces abundant and a few others rare as could be the case without using the coordination mechanism.

### References

- [1] Bittorrent Protocol Specification, <http://www.bittorrent.com/protocol.html>
- [2] B. Cohen, “Incentives build robustness in bittorrent,” in *P2P Economics Workshop*, Berkeley, CA, 2003.