
**BITTORRENT PROTOCOL AND ITS APPLICATION IN CLOUD COMPUTING
ENVIRONMENT.**

DEVIKA K.H,HARITHA K.R,AMRITHA.C,

STINA C THOMAS,RETHINAM.C

ABSTRACT

BitTorrent is widely used as a protocol for file sharing and has gained popularity due to its scalability and performance advantages. This paper proposes a modification to the BitTorrent protocol for Peer-to-Peer communication . The proposed method has been implemented and performance measured on an experimental set up. Results reveal the performance gains that can be obtained. The paper also discusses how the modified BitTorrent protocol can be used to improve Peer-to-Peer communication in a Cloud computing environment.

KEYWORDS

Peer-to-Peer communication, Network traffic, BitTorrent Protocol, Performance.

1. INTRODUCTION

BitTorrent protocol used for Peer-to-Peer file distribution is receiving a lot of attention in computer networking. It enables fast downloading of large files using minimum Internet bandwidth. In traditional Client/Server architecture the Client communicates only with the Server for all information needs but as the number of Clients increases performance degrades due to increase in the Server Load. BitTorrent protocol overcomes this drawback and maximizes transfer speed by gathering pieces of the file required and downloading these pieces simultaneously from Peers who already have them. This process reduces downloading time of very large files, such as videos and television programs more than that possible with other protocols. The paper proposes modification to BitTorrent protocol to make it still faster by allowing them to download files from other Peers without having the complete file. The paper discusses the BitTorrent protocol in Section 2 and its modifications in Section 3. The modified protocol has been implemented and performance measurements carried out on an experimental set up as described in Section 4. .Section 5 of the paper discusses application of the modified BitTorrent protocol for data transfer in a Cloud environment.

2. Peer-to-Peer Communication using BitTorrent Protocol

Normally, communication in a network is based on Client/Server architecture, where communication between a Client and the Server follows unicast (point-to-point communication) method i.e., the communication takes place only between the Server and Client as shown in Figure 1.

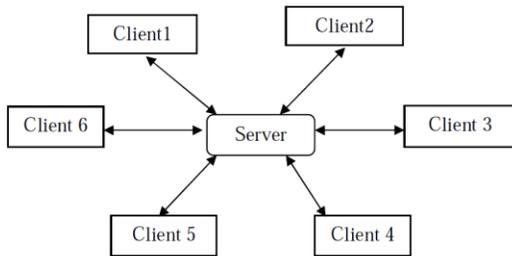


Figure 1. Client/Server Communication

If the network contains a large number of Clients, the Server may fail to respond to the entire Clients' requests due to heavy load on the Server. Moreover when Clients are distributed far and wide, the network traffic increases substantially and becomes a performance bottleneck . As the number of Clients increase, the load at the Server also increases. Inappropriate distribution of resources can lead to increase in traffic resulting in network congestion and possible packet loss. In such cases, Peer-to-Peer communication may be advantageous as all data transfer between Clients need not pass through a central Server.

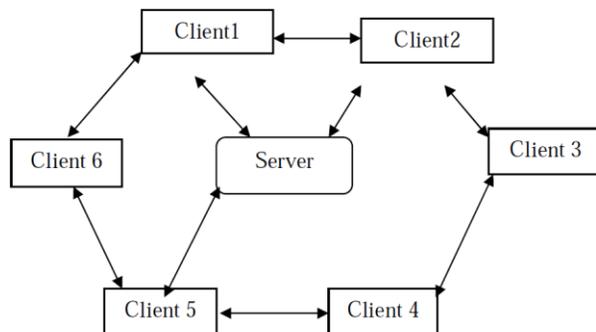


Figure 2. BitTorrent Protocol Scenario

In Peer-to-Peer (P2P) communication, Peers allow its resources, such as processing power, diskstorage, network

bandwidth, and files to be shared and make them available to other network Clients, without the need for any central coordination [11]. Peers behave both like suppliers and consumers of resources. In contrast to the traditional Client–Server model, BitTorrent protocol establishes communication between Peers. BitTorrent is a fast and efficient protocol for distribution of files over a network and uses file sharing technology to transfer large files directly between Clients' computers. This protocol has the advantage of reducing the traffic on Server and on communication links when distributing large files. Rather than downloading a file from a single Server, the BitTorrent protocol allows Clients to join a cluster of Clients to download files from each other simultaneously. This protocol has advantages over all the other protocols proposed for Peer-to-Peer communication as discussed in [4]. BitTorrent protocol allows Clients in a network with low communication bandwidth to distribute files to other recipients more efficiently.

The important entities involved in BitTorrent Protocol are:

- Peers or Seeders (Clients).
- Leecher
- BitTorrent Tracker
- Server

- Torrent File

Peers: These are Clients which are ready to share files with other Clients in the network. They are also called Seeders.

Leechers: These are the nodes in a network who do not want to share its files with other nodes.

BitTorrent Tracker : The Tracker resides in the Server and contains information that include:

1. The IP addresses of the Peers which are connected to the Tracker.

2. The hash number associated with each file segment or piece. (The file is fragmented into number of segments called “pieces”).

3. Hash numbers of the file segments which have already been downloaded from Peers (using SHA-1 Hashing algorithm). This improves security by comparing the hash number of each

segment with the original hash number of the entire file. Tracker keeps track of all the Peers which are connected to Server. Tracker does not allow any unauthorized Client to communicate with Server thereby ensuring security. Tracker also enables the Clients to download files from Server and also from other Clients in the network.

Torrent File: This file contains information about the files to be shared in a network i.e., metadata about the files. This file contains hashes for each segment before sharing it with other Clients. It also contains the information (IP Address) about the Tracker.

3. Modified BitTorrent Protocol

Even though the BitTorrent protocol improves efficiency of a network by sharing files between Peers, it has the disadvantage that a node can share a file with other nodes in a network only when it has the complete file with it, which slows down the performance. To overcome the drawbacks of the BitTorrent protocol it is modified as follows:

1. Incorporating Parallel Downloading: a Client can download segments (of size 1 KB) of a file from a Seeder and from a Server in parallel.
2. Allowing a Peer or Seeder to share a file segment with other Clients in a network even though it does not have the complete file as soon as it downloads the segment from Server.
3. Implementing Centralized Tracker, for enhanced security instead of Decentralized Tracker. In the Decentralized Tracker Peer does not know the identity of the Client authorized to share resource with it and does not provide adequate security.

3.1 Parallel File Sharing using modified BitTorrent Protocol:

The following steps for parallel file sharing using modified BitTorrent protocol is explained for three Clients and a Server:

- Client1 sends a request to the Server for a file. Then torrent file is created for the requested file in the Server and it downloads the torrent file.
- At the same time, if Client2 requests for the same file which is being downloaded by Client1, Client2 starts downloading file segments from both Server and from Client1 in parallel and as Client1 shares its file with Client2 it acts like a Peer or Seeder.
- When Client3 requests for the same file, which is in Client1 and in Client2, it starts downloading simultaneously from Seeder1 (Client1), Server and from Seeder2 (Client2).

4. Test Bench for Performance Analysis

Performance Analysis of the modified BitTorrent Protocol for Peer-to-Peer communication was carried out by implementing on a Test Bench with one Server and three Clients (Client1, Client2, and Client3) connected in a LAN. This Test Bench differs from a normal LAN environment, as Clients do not belong to the same domain. In the setup, a Client can share files with another Client only when sharing between Clients is enabled by the Network Administrator. The experimental setup of Test Bench consisting of Clients (Peers), Tracker, and Server is shown in Figure 3.

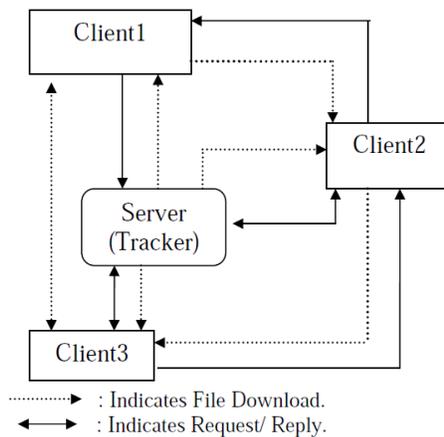


Figure 3. Experimental Set up

Figure 3 shows Client2 and Client3 downloading file segments in parallel from a Server and from other Peers. Initially Client1 gets file from Server and when it is downloading the segments, it can share these segments with other Clients who are requesting for these. Initially Client2 requests for a file from Server, and then Server sends details of the other Peer(s) who are having the same file. Then Client2 starts downloading segments from both the Server and Peer simultaneously. In this set up Client2 gets details of Client1 before it starts downloading segments from Client1 as well as from Server. The same procedure is followed for Client3 which downloads file segments from Client1, Client2 and also from Server in parallel.

5. APPLICATION OF MODIFIED BITTORRENT PROTOCOL IN CLOUD ENVIRONMENT

Although Cloud computing is rapidly emerging as a new paradigm for delivering IT services on “pay -per -use” basis, successful implementation of Cloud computing needs solution to many performance related issues. The important entities involved in Cloud computing are:

- Cloud Server: Cloud Server (CS) or Cloud Service Provider (CSP) which is a pool of resources in a data centre and provides huge processing power and variety of services to the Clients.
- Clients (Peers): They demand services from CS/ CSP. The services may include computing power, storage resources, applications, and processes.

While Cloud computing looks very promising, providing the services from the Cloud Server to the all Clients within a short time continues to be a challenge, as Cloud environment can have a large number of Clients. Performance degradation may occur due to heavy load at Server. To overcome this problem, modified BitTorrent Protocol discussed in this paper can be used to share a large number of files including video, and audio files among peers. When a Client requests for a particular file, the Cloud Service Provider normally provides the requested file, but sometimes CSP may fail to provide requested service due to heavy congestion. In such a situation, Peer-to-Peer communication can be used and Client can seek a file from another Client/Peer instead of from CSP. Taking an entire file from a single Peer can increase load at Peer, hence parallel downloading is preferred, so that, some of the file segments can be downloaded from a Peer and some from CSP in parallel. Download time of each Client can be reduced substantially. Figure 9, shows how Peer-to-Peer Communication works using modified BitTorrent protocol in a Cloud Environment.

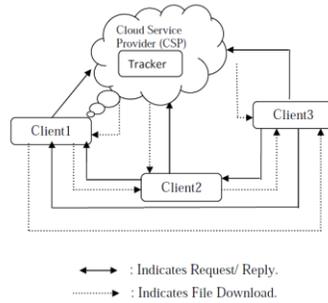


Figure 9. Peer-to-Peer Communication Using Modified BitTorrent in Cloud Platform.

Modified BitTorrent Protocol can be used in Cloud Environment to distribute the Virtual Machine Images efficiently without causing any congestion in the compute nodes. The torrent file is created in the Tracker. A Client/ Peer which needs VM Image starts downloading from the Tracker as well as from other Peers which is having VM Image in Parallel. Hence VM Image can be distributed to the Clients within a short period of time, without substantial increase in network traffic and packet loss.

6. CONCLUSIONS

In this paper, BitTorrent protocol which is widely used in network communication has been discussed and modifications suggested for speeding up file transfer. The modified BitTorrent protocol has been implemented on a test bed and performance analyzed which shows improvement that can be obtained. The paper has also discussed the advantages of using the modified BitTorrent protocol in a Cloud Computing platform to improve the performance.