Variable Chunk Based Parallel Switching To Minimizing File Download Time in P2P Network

Amit Chougule[1], Shambhuraj Deshmukh[2],

[1]Assistant Professor, Bharati Vidyapeeth's College of Engineering, Kolhapur.

[2] Lecturer, VVP'S Institute of Engineering & Technology, Solapur.

Abstract - The Peer-to-peer (P2P) computing has been one of the emerging technologies in distributed file sharing. Experimental studies show that for a file download, service capacity fluctuation takes minutes to several hours. For a P2P one of the fundamental performances metric is the average download time. The common approach to analyse the average download time is average service capacity. Heterogeneity and fluctuation have significant impact on service capacity and hence the averages download time. Random Chunk Based Switching is one of the file downloading scheme where, the file to be downloaded is divided into many chunks but the short comes of this scheme are user sequentially download one chunk at a time, also if user gets stuck in a low service capacity peer, downloading a fixed amount of bytes from that peer may take a long time and here chunk size remains fixed and it does not change with time. We propose a new approach where chunk size changes with time and our model provides no. Of parallel connections which changes with hardware utilization which overcomes the shortcomings of Random Chunk Based Switching[2].

Keyword s- **Service Capacity, P2P Connection, .**

I. INTRODUCTION

The basic idea of P2P network is to have peers participating in file sharing network and a P2P system that is distributed. Every computer (peer) in the network can act as both a server and a client. If a host has a popular file and many peers are requesting it, as soon as one of the downloading hosts finishes the download, [4] it can become a server to service other peers in the network. As time passes on due to the increase in the number of servicing peers the service capacity of the entire network also increases.

Research on P2P system considers traffic measurement, workload analysis and behaviour of peers of P2P applications like Gnutella and Napster and it has been observed that there is significant heterogeneity in peers bandwidth, [4]availability and transfer rate. Much of the research is focused on characterizing overall P2P system e.g. request pattern, traffic volume, traffic categorization.

Several studies have explored how host dynamics within peer-to-peer networks affect performance and reliability. Some researchers have proposed models of web, like a generative model of web traffic on ON-OFF behaviour of web clients and analytical model of file-sharing, to explore the impact of freeloaders on system performance and web system to explore how web caching performance scale with population size.

Earlier results show that a file download session in a P2P network is rather long and varies a lot from user to user. Due to the distributed nature of the P2P network, Searching and locating data of interest in the network has been an important issue in the literature.

If we want to minimize the download time for each user, reducing the actual file transfer time would make a noticeable difference. Most of the recent studies, however, have focused on reducing the total download duration. As the measurement study shows that per-user performance in a P2P network may be even worse than that of centralized network architecture. Those results suggest that there is much room for improvement in the P2P system in terms of per-user performance.

Some researchers have studied the dynamics and structure of P2P e.g. connectivity and host/peer behaviors, properties of shared online content.

Although researchers have made considerable efforts to study the performance of various P2P file sharing application like Bit-torrent, Gnutella, Napster, there is need to study the performance of actual P2P system and performance of individual user.

However, there have been very few results in minimizing the download time for *each user* in a P2P network. In recent work [2], the problem of minimizing the download time is formulated as an optimization problem by maximizing the aggregated service capacity over multiple simultaneous active links (parallel connections) under some global constraints. There are two major issues in this approach. One is that global information of the peers in the network is required, which is not practical in real world. The other is that the analysis is *based on the averaged quantities*, e.g., average capacities of all possible source peers in the network. The approach of using the average service capacity to analyze the average download time has been a common practice in the literature.

A. Service Capacity in P2P Network

The service capacity of a P2P system can be viewed in two regimes. [4]One is the *Transient Phase* in which the system tries to catch up bursty demands. Both analytical model and trace measurements exhibit the exponential growth of service capacity during the transient phase. Second is the *Steady State*, the service capacity of a P2P system will scale and track the offered loads, so individual user's performance will not degrade significantly. Both analysis and empirical data suggest that at higher offered loads and with cooperative users the system performance might improve.

Spatial Heterogeneity in the available service capacities of different source peers and the temporal correlation in the given source peer has significant impact on service capacity.

B. Our contribution

The main contribution of this paper is that, (1) during downloading if we get stuck in a bad source peer with low bandwidth capacity; there arises a need to switch from low bandwidth peer to high bandwidth peer. (2) To design a distributed approach for chunk based switching. (3)To design a Source Replacement approach.

II. FACTORS OF AVERAGE DOWNLOAD TIME

In this section, we consider the heterogeneity of over different network paths and the fluctuation of the capacity over time for a given source peers.

A. Heterogeneity of Service Capacity

In a P2P network, the service capacities from different source peers are different. There are many reasons for this heterogeneity. On each peer side, physical connection speeds at different peers vary over a wide range. On the network side, peers are geographically located over a large area and each logical connection consists of multiple hops. Hence, we assume that those factors mainly determine the long-term average of the service capacity over a given source peers.

B. Correlations in Service Capacity

While the long-term average of the service capacity is mainly governed by topological parameters, the actual service capacity during a typical session is never constant, but always fluctuates over time. There are many factors causing this fluctuation. *First*, the number of connection a source peer allows is changing over time, which creates a fluctuation in the service capacity for each user. *Second*, some user applications running on a source peer, such as games, may throttle the CPU and impact the time amount of capacity it can offer. *Third*, temporary congestion at any link in the network can also reduce the service capacity of all users utilizing that link.

III. EFFECT OF EXISTING METHOD FOR FILE DOWNLOADS IN P2P NETWORK

In this section, we briefly discuss all the existing methods that are used to download a file from P2P networks. And we design our new Distributed Variable Chunk Based Switching. Generally, if a downloader relies on a single source peer for its entire download, then it results in high download time. Since the service capacity of each source peer is different and fluctuates over time, utilizing different source peers either simultaneously within one download session would be a good idea to diversify the risk. The existing methods are (i) Parallel Downloading; (ii) Random chunk-based switching; (iii) Periodic Switching.

A. *Parallel downloading*

Parallel downloading [2] is one of the ways to reduce the download time. In parallel download, a file *F* gets divided into *k* chunks of equal size and single file is allowed to download in parallel with *k* simultaneous connections. Parallel downloading is better than single downloading. In the network with single one user, parallel downloading may not reduce the download time up to the mark. If we can make the chunk-size proportional to the service capacity of each source peer, parallel downloading can produce a better result. But such scheme requires global information of the network

B. *Random chunk based switching*

In the random chunk-based switching [2] scheme, the file to be downloaded is divided into many small chunks. A user downloads the chunks sequentially one at a time. Whenever a user completes a chunk from its current source peer, the user randomly selects a new source peer and connects to it to retrieve a new chunk. In this method, peer, it will stay there for only the amount of time required for finishing one chunk. The download time for one chunk is independent of that of the previous chunk. The switching of source peers based on chunk can reduce the correlation in service capacity between chunks.

However, there is another factor that has negative impact on the average download time, the spatial heterogeneity. The main disadvantage of chunk based switching is that, if we get stuck in a source peer with very low service capacity, downloading a fix amount of bytes from that source peer may take a long time. We can avoid this long wait by making the size of each chunk very small, but this then would cause too much overhead associated with switching to many source peers and integrating those many chunks into a single file.

C. Random Periodic Switching

Instead of waiting to get the complete chunk, we randomly switch [2] between the source peers based on time. There are two schemes in this method

(i) Permanent Connection: when the downloader wishes to download a file, the downloader will choose one of the given source peers randomly with equal probability.

(ii) Random Periodic Switching: here the downloader randomly chooses a source peer at each time slot,

independently of everything else. It is observed that both the spatial heterogeneity and the temporal correlation in the service capacity can significantly increase the average download time of the users in the network.

D. Dynamically Distributed Parallel Periodic Switching

Dynamically Distributed Parallel Periodic Switching (D2PS) that effectively [1] removes correlations in the capacity fluctuation and the heterogeneity in space, thus greatly reducing the average download time. There are two schemes in this method.

1) *Parallel Permanent Connection:* here the downloader randomly chooses multiple 'k' source peers over 'N' possible source peers and it make permanent connection for the fixed time slot 't'.

2) Parallel Random Periodic Switching: here the downloader randomly chooses multiple fixed k source peers over N possible source peers and it makes parallel connection with that k source peers for each randomly selected time slot.

E. Limitation of the Existing methods

Here we find that the existing method has follows shortcoming

1. Average service capacity alone is not sufficient to describe each user's average performance.
2. Source selection function randomly selects source peer but that source may be with low bandwidth.
3. A source has more bandwidth but downloader cannot utilize this bandwidth.
4. Chunk size is fixed for a f ixed time slot and doesn't changes with time.
5. Random chunk based switching is sequential approach.

IV. DISTRIBUTED VARIABLE CHUNK BASED PARALLEL SWITCHING

Here we will see how distributed variable chunk based switching works.

A. System Information

As shown in (Fig.1) every peer in the group contains Data packet, Control Packet and Peer Index. Here we are using two connections (Rendezvous Process) between any two peers. *Data Packet connection:* where only data packets get transmitted between peers. *Control Packet connection*: where only control packets get transmitted between peers. Control packet contains Peer ID, Bandwidth of that peer, and number of source it contain and number of groups it has joined. Control packet helps to decide the source selection and chunk size. Control packets gets exchanged between the peers after a fixed time slot (Fig. 2) and each peer is maintains a Peer Index (Fig.4) where information regarding all peers is maintained from the control packets history. Here Rendezvous process between any two peers uses Rendezvous Protocol which is designed to propagate messages between peers within a group and minimizes network traffic also minimizes burden on the peers and it increases bandwidth

B. *Chunk size decision module:*

After the downloader has received the control packet, we decide the chunk size of the peers (Fig. 6). The downloader will be connected to no. of peers in the group and the downloader will be downloading the file in parallel from these different peers. If bandwidth available is increased then downloading can complete before specified time. If bandwidth available is decreased then downloader will search another peer with good bandwidth and get it replaced. After downloading all chunks from the all sources, the system will check whether the entire file got downloaded or not.

Data Packet	Control Packet	Peer Index

Fig. 1 Peer Diagram

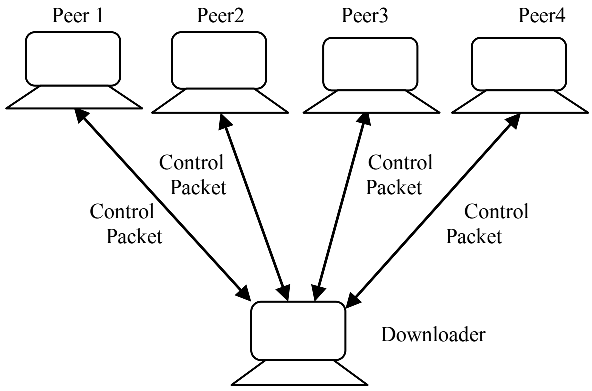


Fig. 2 Control Packet Flow Diagram

Peer ID
Bandwidth
No. of source
No. of Group

Fig. 3 Parameters of Control Packet

C. Parallel Switching

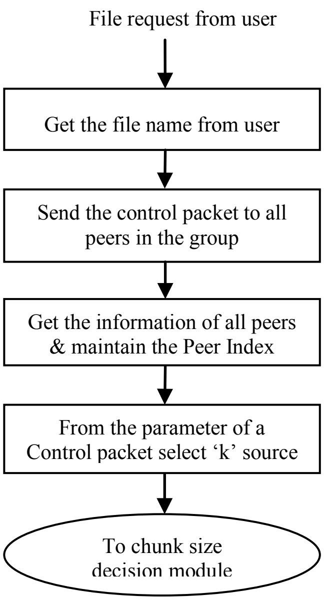
In chunk size decision module we can decide the chunk size, here in Parallel Switching we have provided no. of parallel connection between a downloader an d a source for downloading. The parallel connections are nothing but the threads provided by the downloader or by source. I n P2P network all the participating nodes have different hardware configuration means diff. Memory size, diff. Processor etc., so there CPU utilization is also different. It is seen that total processing power of a node is not fully get utilized so Depending upon the CPU utilization we will increase or decrease the parallel connections between downloader and source for downloading. May be at the time of downloading the CPU utilization get increased so increase the parallel connection but up to certain limit, also it happens that CPU utilization get decreased so decrease the parallel connection but at least one connection is require. E.g. a node have 3GB of memory so that keep 3 parallel connection, if node have 2GB of memory then keep 2 pa rallel connection for downloading

V. CONCLUSION

In this paper we have developed our new scheme Distributed Variable Chunk Based Parallel Switching which removes Heterogeneity and Fluctuation and minimizes the file download time also provide a distributed approach to a sequential Random Chunk Based Switching.

Node1	Bandwidth No. of sources No. of Groups
Node2	Bandwidth No. of sources No. of Groups
Node3	Bandwidth No. of sources No. of Groups
	..
	..
	..
Node n	Bandwidth No. of source No. of Group

Fig. 4 Peer Index Diagram



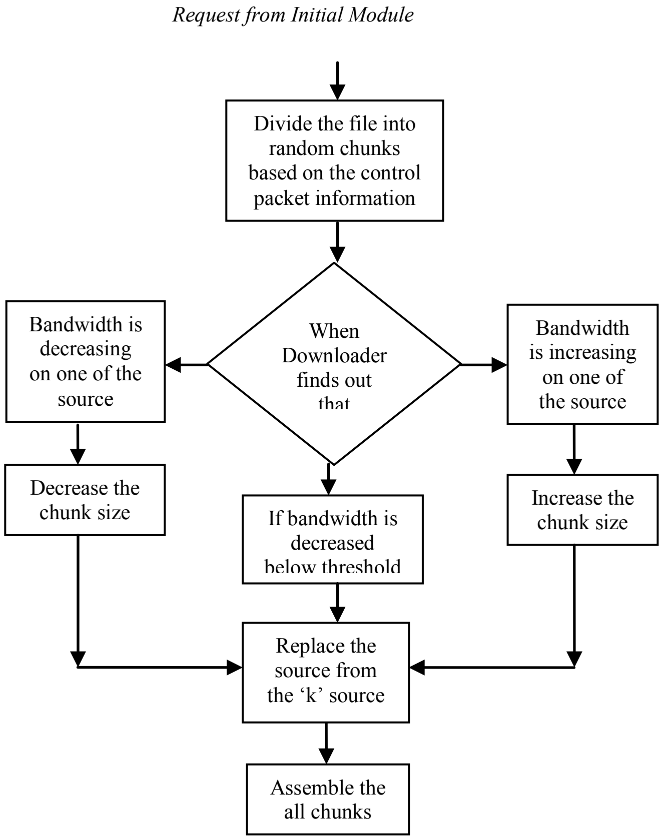


Fig.6 Chunk Size Decision Module

REFERENCES

[1] M. Shyamala Devi, S. Pushpalatha, "Dynamically Distributed Parallel Periodic Switching - D2PS for Minimizing File Download Time in Peer-to-Peer Networks" in 2009 IEEE International Advance Computing Conference (IACC 2009) Patiala, India, 6-7 March 2009

[2] Y. M. Chiu and D. Y. Eun, "Minimizing file download time over stochastic channels in peer-to-peer networks," in IEEE/ACM Transactions on Networking, VOL. 16,NO. 2, APRIL 2008.

[3] "Modelling and performance analysis of Bit-Torrent-like peer-to-peer network", by D. Qiu and R. Srikant in *Proc. IEEE/ACM SIGCOMM*, Aug. 2004.

[4] X. Yang and G. deVeciana, "Service capacity of peer to peer networks," in Proc. IEEE INFOCOM, Mar. 2004, pp. 2242-2252.

[5] K. P. Gummadi, R. J. Dunn, and S. Saroiu, "Measurement, modeling, and analysis of a peer-to-peer file sharing workload," in Proc. ACM Symp. Operating Systems Principles (SOSP), 2003.