Adaptive Sliding Piece Selection Window for BitTorrent Systems

Ahmed B.Zaky ahmed.ahmed02@feng.bu.edu.eg
Shoubra Faculty of Engineering
Electrical Engineering dept/Computer Engineering
Benha University, Egypt

May A.Salama may.mohamed@feng.bu.edu.eg
Shoubra Faculty of Engineering
Electrical Engineering dept/Computer Engineering
Benha University, Egypt

Hala H.Zayed hhelmi@benha-univ.edu.eg
Faculty of Information and Computing
Department of Computer Science
Benha University, Egypt

Abstract

Peer to peer BitTorrent (P2P BT) systems are used for video-on-Demand (VoD) services. Scalability problem could face this system and would cause media servers not to be able to respond to the users' requests on time. Current sliding window methods face problems like waiting for the window pieces to be totally downloaded before sliding to the next pieces and determining the window size that affects the video streaming performance. In this paper, a modification is developed for BT systems to select video files based on sliding window method. Developed system proposes using two sliding windows, High and Low, running simultaneously. Each window collects video pieces based on the user available bandwidth, video bit rate and a parameter that determines media player buffered seconds. System performance is measured and evaluated against other piece selection sliding window methods. Results show that our method outperforms the benchmarked sliding window methods.

Keywords: BitTorrent, Sliding Window, Video-on-Demand, Peer-to-Peer.

1. INTRODUCTION

Video on Demand (VoD) applications are used to deliver different video files to many users. In recent years, VoD applications are widely used for many types of networks with different organizations. Due to the development of computing resources, communications, data storage, and presentations' methods as video websitesa user can select and show any video file from huge sets of videos at any time.

Users can share their resources by using peer-to-peer file sharing or streaming across a network instead of using classical centralized video server[1]. So a limited number of video servers will be required to provide the main service for users. The idea of P2P system is to use computer to computer data transmission not a traditional client server application. P2P applications share different resources as storage, content or any resources available between users. The resources are not centralized but are distributed across the network. P2P is an unpredictable environment [2] for content delivery. However, many P2P file sharing applications were developed such as bitTorrent [3] and FastTrack [4].

BT protocol is one of the most effective mechanisms for P2P content distribution and one of the most popular file transfer protocols over the internet. It is a hybrid P2P system using a dedicated central server for controlling peers' communication. The server building information and the metadata lookups are about the shared contents at each peer node. Metadata is used by a peer's request to identify the nodes having the requested data. Communication channels and data transfers will be handled by the peers not the servers. The main issue is the failure of the central server.

BitTorrent is a file sharing protocol that applies the hybrid mechanism. It is composed of a central server called tracker that stores torrent file containing information about the file shared, as length, hash information and URL of the tracker server[5]. Tracker server has updated information about the peers' list having the requested file and can link the peers to start transferring the pieces of the file to the request issuer.

Many researches were done to build P2P systems for audio and video streaming so that the user can download and upload a video stream in the same time[1]. A P2P systemis scalable, as it will be constructed by users in a network not only by the video server. The video streaming network will be defined by users' locations and network topology that affects the system performance, scalability, fault tolerance and how the system will be maintained [4-6].

BT effectiveness comes from its ability to transfer files quickly by managing peers' bandwidth. Current implementation of BT does not support time based data as video streaming [6]. BT implementation supporting video streaming for P2P systems will add two important features to the users:

1- Minimizethe time required to start the video.
2- Distribute the load of an overloaded server among peers.

Using other network resources, like client upload bandwidth, is one of the main targets for using peer to peer systems. BT uses swarm technique by building a torrent file containing information about the data distributed and how it is split in pieces or chunks. Peers download the chunks from other peers and in the same time upload the requested chunks. Transmitting peers are dynamically connected by a centralized server called tracker server that is used to manage peers, find them, store information about peers' bandwidth usage, and store torrent file information as the number of parts a file is split into.

Peer States is one of the main strong points in BT protocol. Peer has two main states while joining a swarm. These states are known as the choked and interested states.

Choking is a signal that a peer (P1) is not intending to send any data to P2 until un-choked. This could be because the peer is not ready, or not willing to fulfill the requests. Interested means that peer has data that P2 does not have, and wishes to acquire. New connections to peers always begin as choked and not interested. Peers will un-choke connected peers who upload fast but are not interested. If the fast uploading peers subsequently become interested, then the worst up-loader gets choked. It is not possible for every peer to share data with every other connected peer at the same time.

The TCP-Protocol used in BitTorrent to connect to other peers gets easily congested, and performs badly when sharing data over many connections at once. So choking is used to limit that congestion, and to help make sharing faster. Choking is also used to make sharing fairer to all, by ensuring that peers who upload more data faster to others get more data uploaded to them. So the more you upload, the more you can download from other peers. And the less you upload, the more likely you are to be choked.

The main strong point of BT is its ability to select only the best peers to transfer the file. This is done by classifying the peers as seeds and leeches where a seed has the entire file and a leech has only parts of the file. BT has become a popular method but due to this popularity, it is facing some issues as:

• High load on the central tracker due to tracking the status of a huge number of connected peers and updating it.

• Initial sharing delay problem.If a torrent has big-sized pieces; the time taken by a peerto download a piece and tostart sharing that piece could be long.

Scalability is another performance issue facing BitTorrent like VoD systems [1].It is important to make the P2P protocol as efficient as possible, and provide a high initial service capacity (number of seeds). Once the P2P system has reached sufficient seeders to downloaders' ratio, the initial service capacity is not necessary anymore, as the system has become self-sustainable.

The proposed systemin this paper modifies the BT protocol and uses buffered time slots to be transferred between peers.The slots collect the video data using a dynamic adaptive sliding piece selection window.

The paper is organized in the following way. Sections 2 and 3shows piece selection methods and windowing selection algorithms respectively. Section 4 presents the proposed dynamic adaptive window piece selection method.Section 5presents the simulation results and section 6is the conclusion of our work.

2. CURRENTPIECE SELECTION METHODS

Selecting pieces to download in an optimizedorder is very important for the performance. A poor piece selection can result in receiving pieces which are not important for the current playing position.

Many piece selection methods were introduced in bit torrent systems,but which one will be appropriate for media streaming?

2.1 Strict Priority

If a single sub-piece (of a piece P), has been selected, the remaining sub-pieces (of P)will be selected before sub-pieces from any other piece. This does a good job of getting complete pieces as quickly as possible[3]

2.2 Rarest First

Peers download pieces that exist the least first. So the player will make sure that the least common pieces requested exist before starting to download the pieces that are common among peers. These common pieces are left for later.

If a swarm has low number of seeds, rarest first method will be applicable. All peers will be interested in downloading the pieces in the seed and not in any other peers[6]. Rarest pieces mostly will exist in seed and may not exist in any other peers especially at the start of a streaming process. Rarest first handles seeds disconnection by replicating the rarest pieces as quickly as possible, before the peers that have this piece stop uploading or stop responding to other peers or before it gets disconnected from the swarm. Thus the risk of losing any piece is reduced.

2.3 Random First Pieces

Ifa peer downloads a file for the first time, the peer will have nothing to upload. It's important to get a complete piece as quickly as possible. Pieces to be downloaded are selected at random until the first complete piece is assembled. The strategy then changes to rarest first. Rare pieces method would collect rare pieces, regardless of their relevance, so it would download a full piece very slowly.

2.4 Endgame Mode

Sometimes a piece will be requested from a peer with very slow transfer rate. This piece will face delay to finish its download. To prevent that delay, the peersends requests to all peers asking for all the needed sub-pieces. Once a sub-piece arrives to the requesting peer, cancels are sent for that sub-piece to save the bandwidth from being wasted on redundant sends. In practice not much bandwidth is wasted this way, since the Endgame period is very short, and the end of a file is always downloaded quickly.

3. WINDOWING SELECTION ALGORITHMS

BitTorrent has one major drawback when it comes to VoD systems. The peer has to wait for the whole video to download before starting to watch the video, because BitTorrent splits the file to be downloaded into pieces that are downloaded in a non-sequential order using rarest-first method [6].

Peers in different video playing positions are interested in different pieces. BT selection algorithm should be replaced by something else in order to get VoD to work well.

Many researches proposed the rarest-first piece selection only within a small window that slides through the file to be downloaded [7]. In order to implement this kind of “windowing”, only minor changes to BitTorrent clients are needed, and the modified clients are compatible enough to be used for downloading data from standard BitTorrent swarms [6].

Windowing BitTorrent for VoD depends on two factors:

1- The choice of windowing algorithm.
2- The importance of piece and window sizes.

Performance gains can be achieved by carefully optimizing the client behavior in each of the above factors. The selection window methods are divided into:

3.1 Fixed Size Window Algorithm

Window size is measured in number of pieces that will be set to a fixed number[8] and will include both the arrived and non-arrived pieces. Pieces are only requested from within the window.

The beginning point of the window is defined as the first piece that has not yet been downloaded but has not missed its playback deadline.

Fixed size use $w = db/c$ where
d is the playback delay,
b is the video consumption rate
c is the chunk size.

3.2 BiToS

A window will be formed from non-arrived pieces only, which means that the distance of the first and the last piece in the BiToS [6]window can grow large. Pieces can be requested from outside the window according to the available bandwidth.

Pieces are requested from within the window with probability p (a value of p=0.8 was recommendedand from outside the window with probability 1- p) [6].

All pieces of the file must be downloaded including missed pieces, because users need to have the entire file at end of download.

Suppose a piece missed: BiToS will continue to download subsequent pieces by expanding its window to reach the end of the video, whereasfixed size window will stop until the piece is downloaded.

3.3 Stretching Window

Is an extension to BiToS.It uses a maximum number of non-arrived pieces in the window.The maximum window size is calculated by the same formula as that in Fixed Size [7]and it represents the distance between the first and the last piece in the window. Thus Stretching Window has two maximum sizes, BiToS-like maximum number of non-arrived pieces in the window and the fixed-window like represents maximum absolute distance between the first and the last piece in the window that are enforced simultaneously.Pieces are still requested from within the window.

4. THE PROPOSED ALGORITHM

As shown in the previous sliding window methods, window size plays an important factor for the streaming performance. The proposed sliding window method builds two sliding windows that run simultaneously for collecting pieces. Windows are sized dynamically and adapted to the user available bandwidth, video bitrate and a system parameter that determines the media player buffered seconds.The buffered seconds will be of a variable length data and will represent the piece size in the window.

The open question in P2P based VoD system discussed in [5] is to which peer, among all peers that have that piece, should a node send a request for a data piece? Standard BT tracker algorithm returns a list of peers having the requested file. A peer is selected at random from that list. Peers start communicating with each other using chocked and interested methods to share the file.

Selecting such a peer at random has the disadvantage that older peers (nodes which arrived earlier and have large data content) receive more requests from many newly arrived peers. Another disadvantage is that peers will request pieces from the fastest response peers which usually have higher bandwidth than other peers that have the same requested piece.

The proposed selection model will build two lists of peers, classified by the upload bandwidth. Lists will be used in piece selection method. The main purpose is maintaining fairness between the peers and the well distribution for piece requests over network peers such that the peer that has high bandwidth will not be overloaded all the time by other peers' requests. The two lists are:

1. Same or lower speed peers list (SPL):containing peers that have the same speed as the peer requesting the video.
2. Higher speed peers list (HPL):containing peers whose upload bandwidth is higher than the peer requesting the video.

The algorithm developed is shown in
figure **1** andbased on the following:

- Window size will have a variable data length according to the number of pieces in the window.
- Two windows will be created, one for the current pieces requested for Download (RDW), and a second separate window for Missed pieces (MW).
- RDW initial window size will be calculated for each peer session requesting a video by using the following formula.

RDWS = B/N*D

WhereRDWS is the window size measured in number of pieces.
B is peer download bandwidth.
N buffered number of seconds.
D video bitrate/second.

- MW has unlimited size and include all missed parts while the video is downloading.Pieces will be requested in reverse sequential order, trying to collect pieces before their end play time. Missed pieces will be requested from high peer list (HPL).

- RDW and MW will be updated by a timer T = 2* RDWS, updates will take the following actions
    - Move downloaded pieces to the video player buffer.
    - Move missed pieces list from RDW to MW.
    - Calculate RDW size according to current peer bandwidth.
    - Move RDW sliding window to next pieces.
    - The window RDW will slide to the next time frame pieces, after all window pieces' download is complete or timer is exhausted.

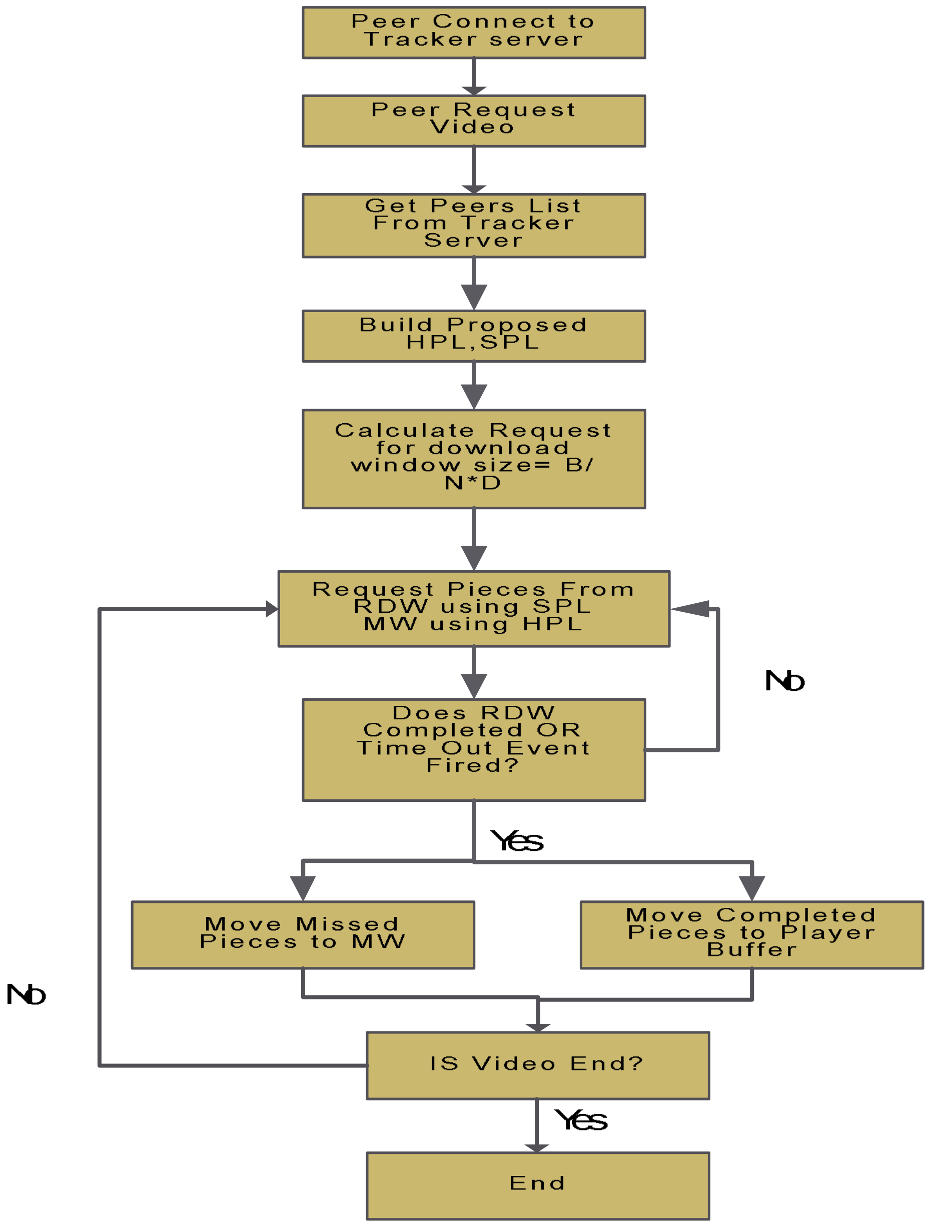
Peer Connect to Tracker server
Peer Request Video
Get Peers List From Tracker Server
Build Proposed HPL,SPL
Calculate Request for download window size= B/ N*D
Request Pieces From RDW using SPL MW using HPL
Does RDW Completed OR Time Out Event Fired?
No
Yes
Move Missed Pieces to MW
Move Completed Pieces to Player Buffer
No
IS Video End?
Yes
End

FIGURE 1: Proposed Piece Selection Method

5. SIMULATION

Simulator results were based on running the proposed method four times on a total number of 167 nodes on a system configuration of 2.5 processor core 2 due, 3 MB cache and 3 GB memory.Event-based P2P simulator called PeerSim[9]has been selected to simulate our model. Simulation parameters used in the simulation of the proposed algorithm are listed in
table **1**.Fixed, BiTos and Stretching windowing methods were also simulated, each ran for 4 times to be compared with the proposed model and with normal BitTorrent.

Parameter	value
Video File size	100 Megabyte
Video Bitrate	512 kb\s
Video Time	26 Minute
Piece size	256 Kbyte
Number of pieces	390 piece
Max swarm size	200 Node
Peer set size	50 Node
Network size	100 Node
Max growth	20 Node
Seeder distribution	10%-20%-30%-40%-50%
Step	10 Seconds
Add	10 Nodes
remove	5 Nodes
Simulation Time	1 Hour

TABLE 1: P2P Network Simulation Parameters

TABLE2shows the number of peers for each bandwidth, which is generated randomly by the simulator in the first time. To be able to measure the system performance and to compare it versus other systems, we force the simulator to work with the same peers' distribution for all other methods (total of 167 peers starting with 100 peers and reaching 167 peers by the end of one hour simulation.

Bandwidth KB/s	Number of Nodes
512	36
1024	49
2048	43
4096	39

TABLE2: Nodes Bandwidth Distribution

The system is tracked for different seeds to peer ratio S/P for 10%, 20%, 30%, 40% and 50%.As shown in figure2,when the number of seeds is 50%, half of the network nodes have the complete video file so the number of pieces transferred in the network is small. For 10% seeds, the piece transfers will be lower, because of the low percentage of seed/peer. When the S/P is 20% the network transfers a number of pieces greater than other distributions. Therefore S/P 20% is selected for the simulation run of all other sliding window methods.The figure also shows the network behavior.Initially, the network piece transfer grows gradually then the transfer becomes stable and with the effect of leaving and entering of new peers with time the number of pieces in the network steps up and down with time.

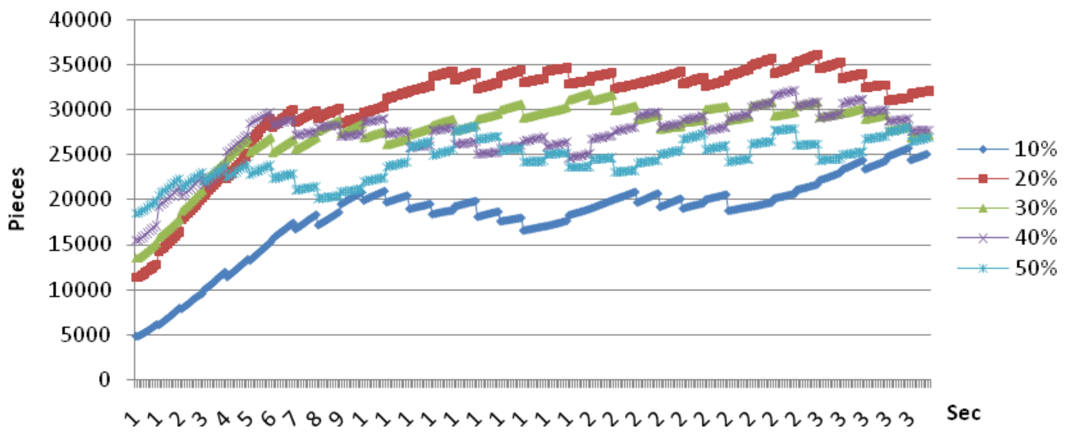

FIGURE2: Proposed Seed Distribution Effect

TABLE3shows the sliding windowing methods running for nodes' bandwidths of 512, 1024, 2048 and 4096. As the difference in bandwidth did not show any significance in the relative performance of the proposed sliding window method compared to others, only the results of 512kb are shown in figure3. Measurements in table 3 are taken for S/P ratio of 20% for all sliding window methods with a random seed distribution

Nodes Bandwidth	Sliding Method	Number of Downloaded pieces	Average joining duration/Min
512	BT	9307	27
	Proposed	8084	34
	Stretch	7438	35
	Bitos	7066	33
	Fixed	6453	34
1024	BT	18822	32
	Proposed	15357	34
	Stretch	14347	31
	Bitos	11255	22
	Fixed	12834	28
2048	BT	8598	10
	Proposed	9670	36
	Stretch	8571	29
	Bitos	3149	3
	Fixed	7031	26
4096	BT	20801	27
	Proposed	14250	34
	Stretch	12184	25
	Bitos	11732	25
	Fixed	11211	35

TABLE3: Proposed Sliding Window Simulation Measurements

FIGURE3 shows that the proposed dynamic adaptive sliding window piece selection method has lower number of pieces compared to the conventional bit torrent selection method. This is normal because BT has no constraints of getting pieces by using rarest first.FIGURE3also shows that the proposed method is superior to other sliding window methods based on bit torrent as it has a greater number of pieces downloaded. FIGURE3 also shows the average joining time for the nodes for each sliding method based on the simulator.

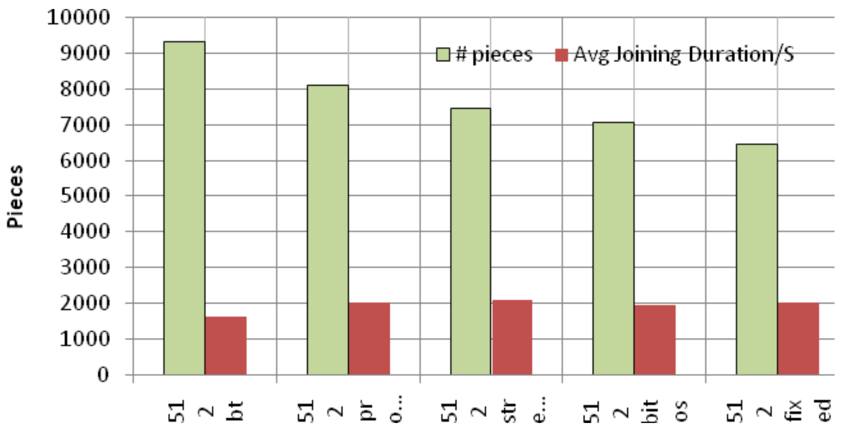

FIGURE3: Sliding Windows Average Usage

FIGURE 4shows the number of pieces summed for all the used bandwidths. It shows that the proposed system is effective for all bandwidthsused.

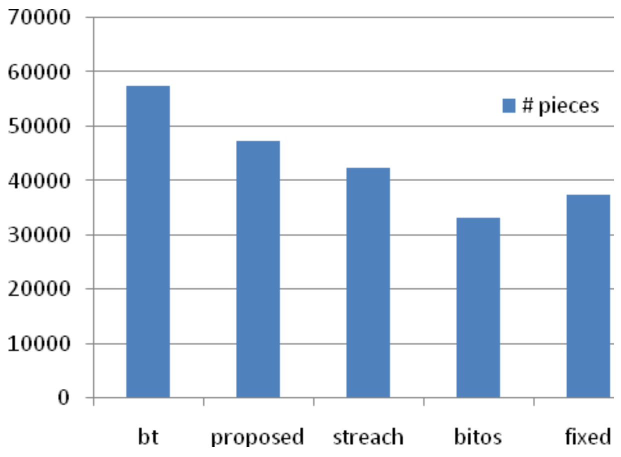

FIGURE 4: Sliding Window Downloaded Pieces

The number of downloaded pieces is not only the critical factor in evaluating the proposed system performance but also the average piece download rate is a critical factoras well.FIGURE 5showsthe average piece download rate for the proposed method withdifferent seeds distribution. It shows that increasing the percentage of seeds generally increases the number of downloaded pieces which is a normal expected performance.

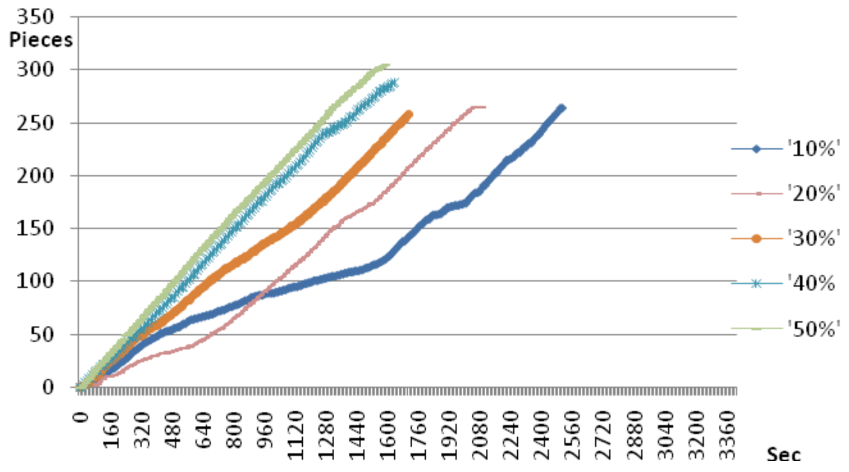

FIGURE 5: Average Piece Download Rate for Different Seeds Distribution

FIGURE 5 can be divided into 2 zones, the first one is from 0 to 900 sec and the second zone is from 900 to the end of the simulation time. In the first zone, the rate of download pieces is random due to random behavior of BT. In the second zone, sliding window method is dominant and the behavior of downloading is as expected i.e increases with the increase in seeds number.

FIGURE 6 shows a comparison of the number of downloaded pieces using the proposed system compared to other methods at a fixed percentage of seeds/peers of 20%. During simulation time the“Stretch” has the least overall performance since it has 2 constraints. It consumes a percentage of bandwidth for downloading out of window pieces as well as it has fixed size window thus limiting the sliding window size.

Bitos comes the second least performance as it has the same first constraint as that of the Stretch. Fixed has the third least performance. Although it has the adaptive capability to adjust the window size, it still has constraints.Firstly, the window size is fixed and secondly, it stops the download if a requested piece has stopped during its download.In the fourth place, just before the BT comes the performance of the proposed algorithm.Since it has no constraints, there ismore flexibility in downloading as the proposed algorithm has two windows: one is open window size as in standard Bittorrent and the other having two lists of peers to select the piecesFigure 6 shows that the proposed method outperforms the 3 methods of Fixed, Bitos and Stretch.It is expected that the proposed algorithm performance is less than that of BT because it uses “rarest first” while downloading the pieces.

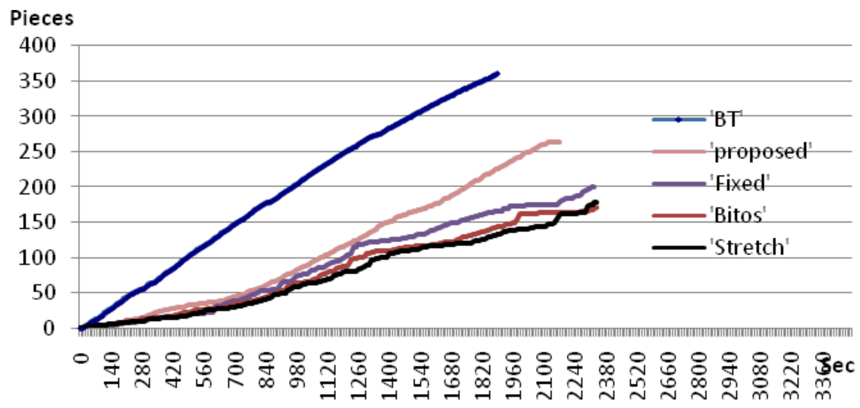

FIGURE 6: Average Piece Download Rate for Different Sliding Window Methods

FIGURE 7show the average downloaded pieces per minute for each bandwidth using different sliding window methods. The measurements are taken after the sliding window networks have become stable in downloading the pieces. Network stability has been achieved after about 5 minutes from running the simulation process as shown in figure 2.

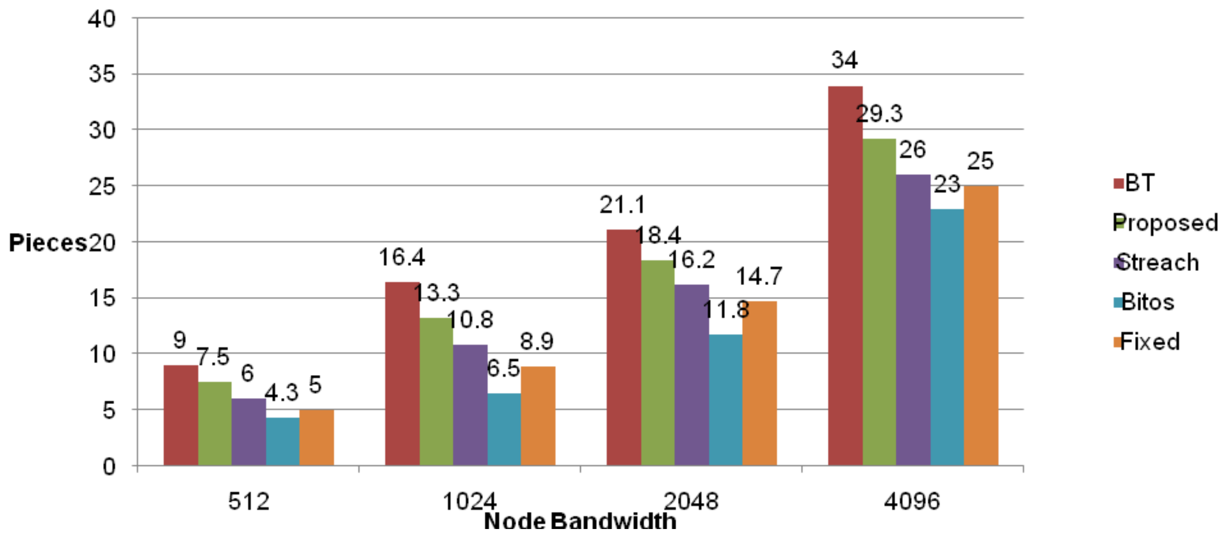

FIGURE 7: Sliding Methods Average Downloaded Piece per Minute

6. CONCLOSION

Due to the deficiency of BT systems in handling timed data, a P2P BT system was proposed to overcome this deficiency. The proposed system presented extensions to the BitTorrent protocol which enhances the video on demand functionality.

The modified BitTorrent protocol is applicable for media streaming by introducing a dynamic adaptive sliding window for selecting the pieces of the video file in faster and in order within a frame of a window buffered seconds.

The simulation results show that the system can efficiently reduce the streaming time and allow more file pieces to be downloaded under the constraint of a sliding window mechanism. The results outperformed the famous window sliding techniques thus proving the efficiency of our proposed system.

7. REFRENCES

1.L.D'Acunto, T.Vinko, andJ.Pouwelse. "Do BitTorrent-like VoD Systems Scale under Flash-Crowds?" IEEE Tenth International Conference on Peer-to-Peer Computing (P2P), 2010, pp 1-4.

2.C.Shirky. "What Is P2P…And What Isn't?".Internet:http://openp2p.com/pub/a/p2p/2000/11/24/shirky1-whatisp2p.html, Nov. 24, 2000[Mar.1, 2011].

3.B.Cohen."The BitTorrent Protocol Specification". Internet:http://www.bittorrent.org/beps/bep_0003.html, Jan.10, 2008[Mar.1 2011].

4."The FastTrack Protocol". Internet: http://cvs.berlios.de/cgi-bin/viewcvs.cgi/gift-fasttrack/giFT-FastTrack/PROTOCOL, July.7, 2004 [jan.1 2010].

5.Y.Yang, A.Chow, L.Golubchik andD.Bragg.“Improving QoS in BitTorrent-like VoD Systems”. In Proceedings of IEEE INFOCOM, 2010, pp 1-9.

6.A.Vlavianos, M.Iliofotou and M.Faloutsos. “BiToS:Enhancing BitTorrent for Supporting Streaming Applications”.25th IEEE International Conference on Computer Communications, April 2006.pp 1-7

7.S.Savolainen, P.Raatikainen,and N.Tarkoma.“Windowing BitTorrent for Video-on-Demand: Not All is Lost with Tit-for-Tat”.IEEE Global Telecommunications Conference (GLOBECOM), 2008. pp 1-8.

8.J.Shah, P.Paris.”Peer-to-Peer Multimedia Streaming Using BitTorrent”. IEEE International Performance, Computing, and Communications Conference, 2007,pp 340-347.

9.M.Jelasity, A.Montresor,G.Paolo and S.Voulgaris.“PeerSim: A Peer-to-Peer Simulator”. Internet: http://peersim.sourceforge.net. 2009 [Mar.1, 2011].

10. F.Frioli,M.Pedrolli“A BitTorrent module for Peersim”. Internethttp://peersim.sourceforge.net/code/bittorrent.tar.gz. 2008 [Mar.1, 2011].

11. S.Saroiu,P.Krishna, D.Steven and D.Gribble.”A Measurement Study of Peer-to-Peer File Sharing Systems”in Proceedings of Multimedia Computing and Networking (MMCN), 2002.

12. B.Chenga, X.Liub, Z.Zhangb and H.Jina. ”*Evaluation and optimization of a peer-to-peer video-on-demand system*”.Journal of Systems Architecture,Vol. 54, pp. 651-663, 2008.

13. S.Keong, L.Crowcroft, J.Pias, M.Sharma and R.Lim. ”*A Survey and Comparison of Peer to Peer Overlay Network Schemes*” IEEE Communications Surveys & Tutorials, Vol. 7, pp. 72-93, 2005.