ISSN 2348 – 7968

Effect of Content Size on Start and Finish Time of P2P based Video on Demand (VoD) Services

Sudipta Majumder[1], Md. Anwar Hussain[2]

[1]Dept. of CSE, DUIET,Dibrugarh University,
Assam, India,Email:sudipta2020@gmail.com

[2]Dept. of ECE, NERIST,
Nirjuli,Arunachal Pradesh, India,Email:bubuli_99@yahoo.com

Abstract

Video on demand (VoD) system is one of the prominent content distribution mechanisms as it provides lots of conveniences and choices to the users. By using Video on Demand services, Users can watch Video as per their requirement and wishes asynchronously. But, the challenges for implementation of such system are not limited. The challenges include wait-time for starting of the service, download time of the whole video content to provide continuous & unhindered services etc. In this paper we have studied the relation between download time of first chunk and the size of whole content if the bandwidth is limited in perspective of fixed upload rate in P2P based Video on demand system. Here, we find out at how does content size of media to be distributed affects the start time video on demand services. The start time of the service depend on the download time of the first chunk. Hence we try to study the relationship between first chunk download time and the content size of the media.

Keywords- Video on demand (VoD), Peers, trackers, Bittorrent, seeder, chocking, Upload data rate, chunk, Download time.

1. Introduction

Peer-to-peer (**P2P**) computing or networking incorporates the distributed application architecture where it partitions tasks or workloads between peers. All the Peers in the peer to peer network have the same capability and rights. The peers are also some times called as nodes. These nodes are assumed to form a peer-to-peer network of nodes.

Peers make a portion of their resources directly available to other network participants without the need for central administration by servers or stable hosts, the resources includes processing power, computing power or network bandwidth etc. . Peers are both servers and consumers of resources, in comparison to the traditional client-server model in which the consumption and supply of resources is divided In this paper we have studied the relation between download time of first chunk and the size of whole content if the bandwidth is limited in perspective of fixed upload rate in P2P based Video on demand system. Here, we find out at how does content size of media to be distributed affects the start time video on demand services. The start time of the service depend on the download time of the first chunk. Hence we try to study the relationship between first chunk download time and the content size of the media. For this we have created closed simulation environment Scenario having 10 peers participating in it and size of Video chunk considered is 120MB to 240 MB in steps of 20 MB.

Peer-to-peer file sharing became popular in 1999 with the introduction of Napster, a file sharing application and a set of central servers that linked people who had files with those who requested files. The central index server indexed the users and their shared content. When a node is searching for a particular segment of the file, the server searches all available copies of that file and return a list of them to the searching node. Here the transfer of the searched file takes place between the two private computers. A limitation was that only music files could be shared [2]. Because this process occurred on a central server, however, Napster was held liable for copyright infringement and shut down in July 2001. It later reopened as a pay service [1].After Napster was shut down, the most popular peer-to-peer services were Gnutella and Kazaa. These services also allowed users to download files other than music, such as videos, movies and games.

Some of the successful peer to peer based content delivery systems are Gnutella [4],KaZaA [5], eDonkey [3], and BitTorrent [10]. These p2p systems are found to be very popular among the user. These systems are generally used to distribute media files such as video files. But the disadvantage of the systems is that the whole video has to be downloaded before the user start

ISSN 2348 – 7968

watching them. To overcome such a disadvantage Cool streaming [13] and its derivate ([4], [6]) have come up with system that can be used to stream live media contents. But the question of recent years is that if peer to peer based systems can be used to provide a near-Video-on-Demand (VoD) service to the users.

A near-Video-on-Demand (VoD) has got a vital advantage which is that the video can be watched while it's being downloaded and besides this, another advantage that it can be watched whenever we want after it has been downloaded and operations like rewind and fast-forward on the video file makes it a very desirable and useful service.

In this paper, we investigate the potential of leveraging P2P networks for a VoD service and provide guidance, and designs principles to efficiently build such systems. Video distribution over the Internet has been one of the most prolific areas of research [8], [9], [15], [22]. The particular problem of designing a near-VoD service has also received extensive attention in the past [7], [11], [14], [15], [16]. An important requirement of a near-VoD service is to be able to support a large number of users, and, hence, such systems should be scalable. The need for scalability becomes clear if we consider that a typical video stream incurs a heavy burden both on the network and the system resources (e.g. disk I/O) of the server. The multicasting paradigm has been proposed to address the scalability issues [7], [12], [15]. Indeed, many systems such as Pyramid Broadcasting [15] and Skyscraper [12] can provide scalable Near-VoD service by using elegant techniques for dividing the video into segments and broadcasting each segment in a different multicast channel. However, both these systems require a multicast-enabled infrastructure.

2. P2P video on demand

The Main Advantages of Video-on-demand (VoD) service systems provide multimedia services offer more flexibility and convenience to users by allowing them to watch any kind of video at any point in time. Such Video-on-demand (VoD) systems are made in such a way that capable of delivering the requested Video cautiously to the user. Unlike live streaming, there is complete control over the media, in VoD systems, with operations such as pause, forward and backward functionalities. The desired behavior of VoD systems is that it can handle large number of video demands made by users asynchronously for watching different parts of the same video at any given time. Implementation of VoD system having such behavior is very challenging specially in tree- based p2p systems because the users will get video exactly in the same order it left the root node as they receive contents directly from the source server. But the Mesh-based P2P systems can handle such situation efficiently as they are able to distribute large files. In this kind of systems a large video is usually broken into many small blocks of pieces. Both the system throughput and the rate, at which the content can be distributed to users, greatly depend on the diversity of the blocks contained at different peers. The challenge of providing VoD services using mesh-based P2P networks lies in the fact that the blocks have to be received at the peer-side in a sequential order, and time constraints have to be considered at all times to guarantee continuous visualization. Hence VoD services using mesh-based P2P can effectively address all the issues related to the efficient Video-on-demand (VoD) services. In the next sub- section we are going to discuss tree based P2P based VoD and Mesh Based VoD to little extent.

2.1 Tree-Based P2p Vod

One of the first IP multicast policies introduced for supporting VoD services was patching [24]. It inspired P2Cast [25] design for distributing video content among asynchronous users. Here each user act as a server for the video content that it has, while receiving portions of videos from other users. Users joining the P2P network within certain time limit form part of a session. Along with the source server, users belonging to the same session form an ALM tree, known as the base tree. Then, the entire video is streamed from the source server using this base tree. Users joining the session join the base tree and retrieve the base stream from it. In addition, new clients missing the initial part of the video must obtain a get the video directly from the concerned server or other users who have already cached the required video. Users behave just like peers in a P2P network.

22. Mesh-based P2P-VoD

In a mesh overlay network, each node contacts a subset of neighbors to receive a number of chunks. Each node needs to know which chunks are owned by its neighbors and explicitly pulls the chunks it needs. In essence, every node relies on multiple neighbors to retrieve the content which makes the system resilient to node-failures. It is worth noting that a peer may simultaneously request multiple neighbors in two ways, namely for either the same content or for different ones [12]. The Users should receive blocks sequentially in order to watch the movie while downloading [19]. Additionally, the nature of VoD systems should ensure the availability of different blocks of video file at any given time, especially if users expect to perform VCR operations during playback [20].

3. Implementing of Simulation Scenarios

We have simulated the scenarios for P2P based BitTorrent Video on demand in NS2. NS 2 is a discrete event simulator. The scalability and robustness of the simulator have attracted us for choosing the video on demand simulation platform. Followings are the list of BitTorrent VoD parameters are used for simulation.

```
// inform TCL script when peer finishes download
#define BT_TELL_TCL

// choking algorithm used in the simulation
#define BITTORRENT 0
#define
BITTORRENT_WITH_PURE_RAREST_FIRST 1

// rolling average (multiple of
CHOKING_INTERVAL)
// In BT: 20s
#define ROLLING_AVERAGE                    2

// time interval between two optimistic unchokes
(multiple of CHOKING_INTERVAL)
// In BT: 30s
#define OPTIMISTIC_UNCHOKE_INTERVAL        3

// time interval between remote peer has to upload at
least one piece to be not snubbed
// In BT: 60s
#define ANTI_SNUBBING                      6

// number of missing pieces to switch from normal
mode to end-game
#define END_GAME                     1

// time to wait before requesting more peers (default =
300)
#define REREQUEST_INTERVAL 300

// time to wait between checking if any connections
have timed out (defaults to 300.0)
#define TIMEOUT_CHECK_INTERVAL   300

// number of seconds to pause between sending keep
alives (defaults to 120.0)
#define KEEPALIVE_INTERVAL 120
```

The test bed for the simulation includes user entered number of peers. Besides this, the test bed allows us desired number of seeders. The upload rate for all the peers can be determined at the run time. Here all the peers participating in the simulation form a neighborhood. A peer can download the desired block from the neighbor if it has the block or it shares the video. Here a tracker is used to keep track of all the peers and the blocks of video file

A Simulation parameters

Table 1: Simulation Scenario 1

Parameter	Value
No of Peers (N_P)	10
No of Seeds (N_S)	1
File Size (S_F_MB)	120 MB
Chunk Size (S_C)	256 KB
No. of Chunks (N_C)	400
Choking Algorithm	Pure Rarest First
Keep alive interval	120 Sec.
Time out check interval	300 Sec
Peer Registered at tracker	10
Upload rate	500 Kbps,

Table 2: Simulation Scenario 2

Parameter	Value
No of Peers (N_P)	10
No of Seeds (N_S)	1
File Size (S_F_MB)	140 MB
Chunk Size (S_C)	256 KB
No. of Chunks (N_C)	400
Choking Algorithm	Pure Rarest First
Keep alive interval	120 Sec.
Time out check interval	300 Sec
Peer Registered at tracker	10
Upload rate	500 Kbps,

Table 3: Simulation Scenario 3

Parameter	Value
No of Peers (N_P)	10
No of Seeds (N_S)	2

ISSN 2348 – 7968

File Size (S_F_MB)	160 MB
Chunk Size (S_C)	256 KB
No. of Chunks (N_C)	400
Choking Algorithm	Pure Rarest First
Keep alive interval	120 Sec.
Time out check interval	300 Sec
Peer Registered at tracker	10
Upload rate	500 Kbps,

Table 4: Simulation Scenario 4

Parameter	Value
No of Peers (N_P)	10
No of Seeds (N_S)	2
File Size (S_F_MB)	180 MB
Chunk Size (S_C)	256 KB
No. of Chunks (N_C)	400
Choking Algorithm	Pure Rarest First
Keep alive interval	120 Sec.
Time out check interval	300 Sec
Peer Registered at tracker	10
Upload rate	500 Kbps,

Table 5: Simulation Scenario 5

Parameter	Value
No of Peers (N_P)	10
No of Seeds (N_S)	3
File Size (S_F_MB)	200 MB
Chunk Size (S_C)	256 KB
No. of Chunks (N_C)	400
Choking Algorithm	Pure Rarest First
Keep alive interval	120 Sec.
Time out check interval	300 Sec
Peer Registered at tracker	10
Upload rate	500 Kbps

Table 6: Simulation Scenario 6

Parameter	Value
No of Peers (N_P)	10
No of Seeds (N_S)	3
File Size (S_F_MB)	220 MB
Chunk Size (S_C)	256 KB
No. of Chunks (N_C)	400
Choking Algorithm	Pure Rarest First
Keep alive interval	120 Sec.
Time out check interval	300 Sec
Peer Registered at tracker	10
Upload rate	500 Kbps

Table 7: Simulation Scenario 7

Parameter	Value
No of Peers (N_P)	10
No of Seeds (N_S)	3
File Size (S_F_MB)	240 MB
Chunk Size (S_C)	256 KB
No. of Chunks (N_C)	400
Choking Algorithm	Pure Rarest First
Keep alive interval	120 Sec.
Time out check interval	300 Sec
Peer Registered at tracker	10
Upload rate	500 Kbps

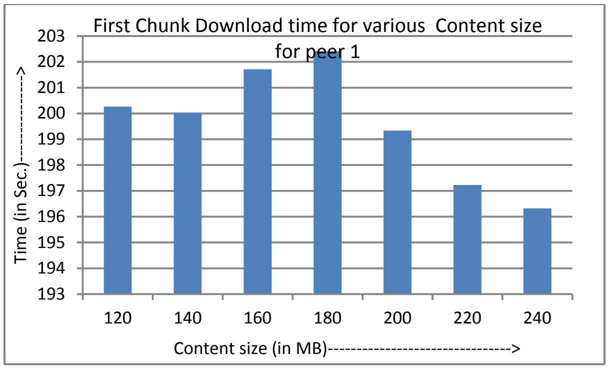

Figure 1: First Chunk Download time for various Content sizes for peer 1 [1 Seed]

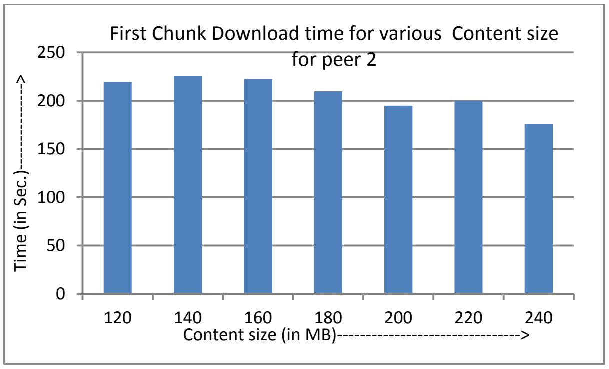

Figure 2: First Chunk Download time for various Content sizes for peer 2 [1 Seed]

Figure 1, 2 and 3 shows the relation between first chunk download time for various content sizes. The size of the contents varies from 120 MB to 240 MB in steps of 20 MB. On careful study of the graph, we notice that the first chunk download time for the peers decreases with the increase of the content size with only few exceptions. We have obtained the graphs for first chunk download time for all the peers but have shown a few graphs for limiting the length of the paper. In the figure 4, we can clearly notice that the first chunk download time for content size 120 MB is greater than that of 240 MB for all the peers. But the decrease in the first chunk download time is not much since it is small network of 10 peer participating in the VoD services. Down load time of the first chunk is important in p2p based video on demand services because it determines the wait time of the viewing the video or media. The wait time is critical quantity because if it is very high, it will affect the consumer as nobody has much time waiting for a video nowadays until and unless it is very urgent. The rest of the chunk of the video will be downloaded while the consumer is watching the first chunk. They will not require to wait for next chunk to be downloaded.

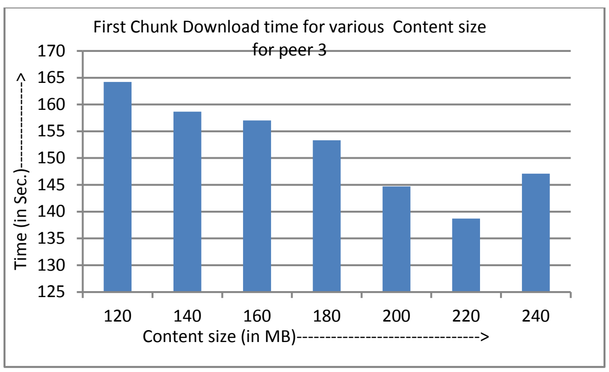

Figure 3: First Chunk Download time for various Content size for peer 3 [1 Seed]

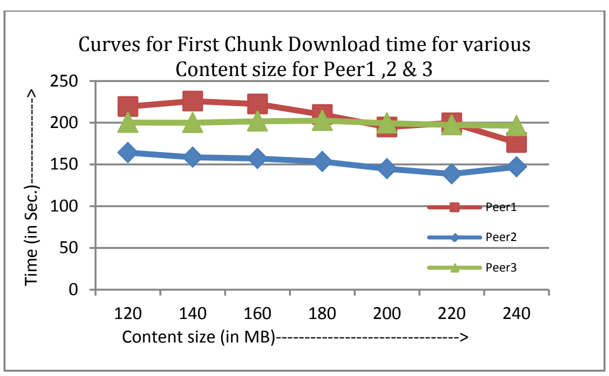

Figure 4: Curves for First Chunk Download time for various Content sizes for peers

Similarly, figure 5, 6 &7 shows the total download time of the media content of various sizes. Here the time taken to download the video content increases even if the video is chopped into small pieces and shared among the peers. The download time of the media content of bigger size takes larger times as compared to that of small sizes. This data set is chosen from hundreds of simulated scenarios with varying upload data rate. It is observed that for the given simulation scenario with varying media content sizes, the time taken to download each and every chunks is most likely to limited to a small range of time. That is, the time taken to down load the chunks won't vary very drastically which is one of the desired characteristics for a p2p based video on demand system. Also the figures 8 tells us the relation between peers' total media download time and media content size.

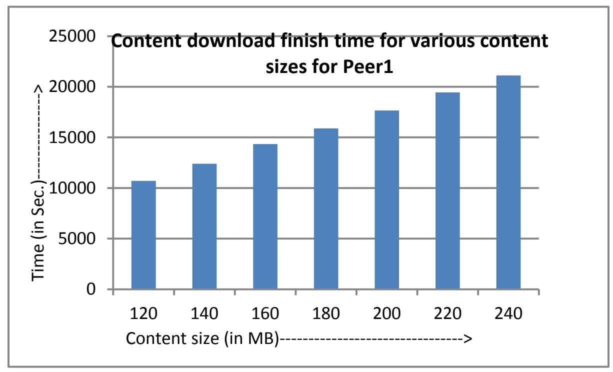

Figure 5: Content download finish time for various content sizes for Peer1

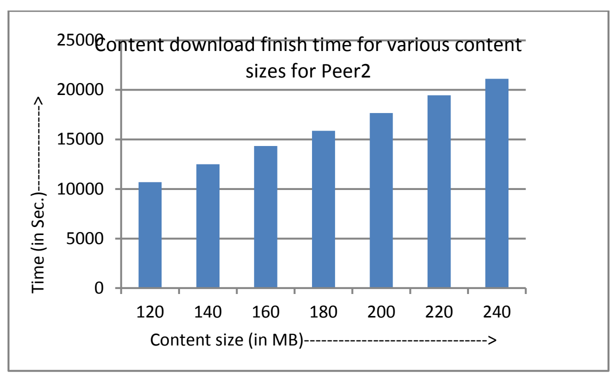

Figure 6: Content download finish time for various content sizes for Peer2

Here A seed refers to a machine possessing some part of the data. A peer or downloader becomes a seed when it starts uploading the already downloaded content for

IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 7, July 2015.
www.ijiset.com
ISSN 2348 – 7968

other peers to download from. This includes any peer possessing 100% of the data or a web seed. When a downloader starts uploading content, the peer becomes a seed. The Seed for all the simulation is kept as one.

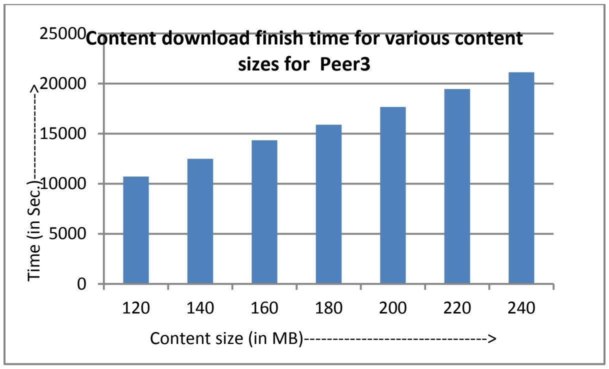

Figure 7: Content download finish time for various content sizes for Peer3

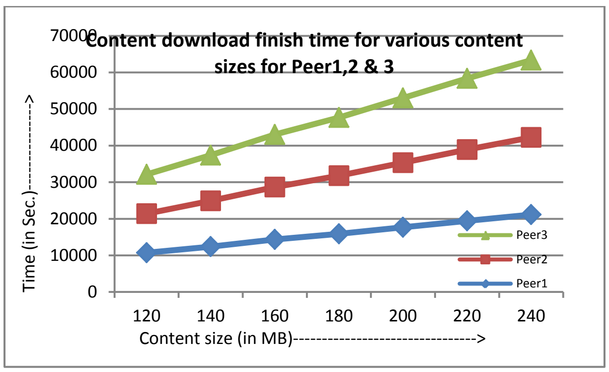

Figure 8: Content download finish time for various content sizes for Peers

Seeding refers to leaving a peer's connection available for downloaders to download from. Normally, a peer should seed more data than download. However, whether to seed or not, or how much to seed, depends on the availability of downloaders and the choice of the peer at the seeding end.

Table 1: Average chunk download time and total download time for various content sizes

Content size(in MB)	120	140	160	180	200	220	240
1st chunk download time(in sec)	155	155	155	154	151	150	151
Total download time(in sec)	9637	11229	12897	14295	15887	17504	19005

The Table 1 above shows the Average 1st chunk download time and total download time for various content sizes. Here the 1st chunk download time is the average for of all the peers in a simulation. Similarly, the total download time is the average of all the total download times for peers in a simulation. Figure 9 is the bar chart representation of table 1. Here we can barely notice the 1st chunk download time in comparison to total download time. On studying the table 1, we can say the fist chunk download time varies a bit for all the peers with for various sized video content to be downloaded.

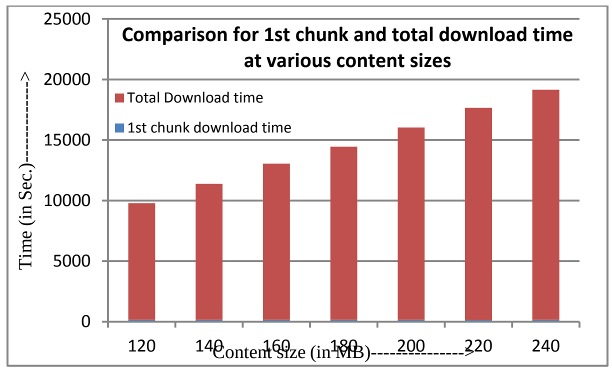

Figure 9: Comparison for 1st chunk and total download time at various content sizes

But the total download time varies significantly for each peer for downloading media content of varying sizes.

4. Conclusion

In this paper, we have studied the relation between content size with Start and Finish Time of P2P based Video on Demand Services. It has been observed that as the size of the content increases, the download time of the first chunk decreases slightly. Because of which the start time of the video also effected slightly. Thus we can conclude here is that the size of the content does not affect the start time of the video on demand services much. That is, the services of video on demand (VoD) starts without much hindrance due to content size. As the total download time is increased greatly, the quality of the service will be affected due to large content size

ISSN 2348 – 7968

since it takes longer time to download the video resulting poor quality of service.

Reference:

[1] Carmack,Carman "How Bit Torrent Works".

[2] Tyson, Jeff.. "How the Old Napster Worked".

[3]eDonkey 2000 - overnet. http://www.edonkey2000.com/.

[4] Gnutella.p://p2pjournal.com/main/gnutella.htm.

[5] KaZaA. http://www.kazaa.com/.

[6] Pplive. http://www.pplive.com/.

[7] Kevin C. Almeroth and Mostafa H. Ammar. On the use of multicast delivery to provide a scalable and interactive Video-on-Demand service. Journal of Selected Areas in Communications,14(6):1110–1122, 1996.

[8] John G. Apostolopoulos, Wai tian Tan, and Susie J.Wee. Video streaming: Concepts, algorithms, and systems.http://www.hpl.hp.com/techreports/2002/HPL-2002-260.pdf, Sep 2002.

[9] Shanwei Cen, Calton Pu, Richard Staehli, Crispin Cowan, and Jonathan Walpole. A distributed real time MPEG video audio player. In NOSSDAV, 1995.

[10]BramCohen. BitTorrent. http://www.bittorrent.com.

[11]Martin Reisslein Despina Saparilla, Keith Ross. Periodic broad-casting with vbr-encoded video. In IEEE Infocom.IEEE Press, 1999.

[12]Kien A. Hua and Simon Sheu. Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems. In ACM SIGCOMM, pages 89–100. ACM Press, 1997.

[13] V.P. Kompella, J.C. Pasquale, and G.C Polyzos. Multicasting for multimedia applications. In IEEE Infocom'92, volume 3, pages 2078–2085. IEEE Press, May 1992.

[14]Simon Sheu, Kien Hua, and Wallapak Tavanapog. Chaining: A generalized batching technique for video-on-demand systems. In International Conference on Multimedia Computing and Systems (ICMCS'97). IEEE Press, 1997.

[15]S. Viswanathan and T. Imiehnski. Pyramid broadcasting for video on demand service. In IEEE Multimedia Computing and Networking Conference. IEEE Press, 1995.

[16] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum. CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media stream-ing. In IEEE Infocom. IEEE Press, 2005

[17] EMULE. Emule Homepage. http://www.emule-project.net/.

[18]Guo, Y., Suh, K., Kurose, J., & Towsley, D. (2003). "P2Cast: Peer-to-peer patching scheme for VoD service". Proceedings of the 12th International Conference on World Wide Web, 301-309.

[19]Annapureddy, S., Guha, S., Gkantsidis,C., Gunawardena, D., & Rodriguez,P. R. (2007). Is high-quality vod feasible using P2P swarming?Proceedings of the 16th International Conference on World Wide Web, 903-912.

[20] Hareesh.K and Manjaiah D.H " Peer-to-peer live streaming and video on demand design issues and its challenges", IJP2P,Vol.2, No.4, October 2011

[21] Kolja Eger, Tobias Hoßfeld, Andreas Binzenhofer, Gerald Kunzmann"Efficient Simulation of Large-Scale P2P Networks:Packet-level vs. Flow-level Simulations"Proceeding UPGRADE '07 Pages 9 - 16 ACM New York, NY,USA.

[22]A. Betker, C. Gerlach, R. H¨ulsermann, M. J¨ager,M. Barry, S. Bodamer, J. Sp¨ath, C. Gauger, andM. K¨ohn.Reference transport network scenarios.MultiTeraNet Report, 2003.

Bibliography

Sudipta Majumder: Mr. Sudipta Majumder is a faculty in the Department of computer science and engineering, DUIET at Dibrugarh University, Assam. He has completed his Masters of Technology from NERIST, Arunachal pradesh in year of 2011.Hs graduation (B.Tech) is from Computer Science & Engineering. His area of interest is Network security.

Md. Anwar Hussain: Prof. Md. Anwar Hussain is a professor in the Department of Electronics and communication Engineering in NERIST, Arunachal Pradesh. His area of interest is wired and wireless communication and networking.