TU Delft

Delft University of Technology

Faculty of Electrical Engineering, Mathematics and Computer Science
Network Architectures and Services

Analyzing P2PTV Traffic Via Measurement

C Jiang
(1379844)

Committee members:

Supervisor:	Prof.dr.ir. Piet Van Mieghem
Mentor:	Siyu Tang
	Dr.ir. Fernando Kuipers
Member:	Prof.dr.ir. Piet Van Mieghem
	Dr.ir. Fernando Kuipers
	Dr. Christian. Doerr
	Dr.ir. Anthony Lo

10th July, 2009
M.Sc. Thesis No: PVM 2009 – 056

Analyzing P2PTV Traffic Via Measurement

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

In

EMBEDDED SYSTEMS

By

Chengcheng Jiang
born in Hangzhou, China

Network Architectures and Services Group
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Preface

This research work was carried out at the Faculty of Electrical Engineering Mathematics and Computer Sciences with the Network Architectures and Services Group of Delft University of Technology. It was from September 2008 to July 2009.

Many people contributed to this work. First I would like to express my sincere appreciation to Siyu Tang, who not only guided my thesis but also encouraged me through the whole process. And I also appreciate the supervising and support from assistant professor Fernando A. Kuipers.

My deepest gratitude goes to my supervisor Prof. P. F. A. Van Mieghem for his precise attitude on science.

I gratefully acknowledge my friends at the Department of Embedded Systems and Telecommunications for making such a productive work environment.

Last but not least, I would like to thank my family for their constant love and support.

Abstract

In recent year, the P2P television (P2PTV) becomes the most popular application for Internet users, due to the rapid development of peer-to-peer (P2P) technology. The efficiency and quality is the most important issue for P2PTV applications. Therefore, understanding and analyzing the performance of different P2P networks is one of the most important aspects in the research field. In this thesis, we evaluate the traffic pattern of a mesh based P2PTV network, i.e. SopCast, via an experimental study. We also try to dissect the operational mechanisms in the SopCast, e.g. criteria of selecting peers, in which way the network copes with node churn, etc. Our experiments are conducted on PlanetLab. A source provider (SP) and a set of PlanetLab nodes are controlled to form a private P2PTV network which makes it possible to collect the traffic of a complete P2PTV network We designed two scenarios during the experiment. The first scenario is a relative stable network, and the second one focuses on the overlay dynamics. We found that the super-peers selection algorithm of SopCast is based on the registration time with the SP, and the SopCast system copes well with the dynamics of super-peers. The results help us to understand the behavior and the operational mechanism of SopCast. The analysis also helps us to understand similar P2PTV networks.

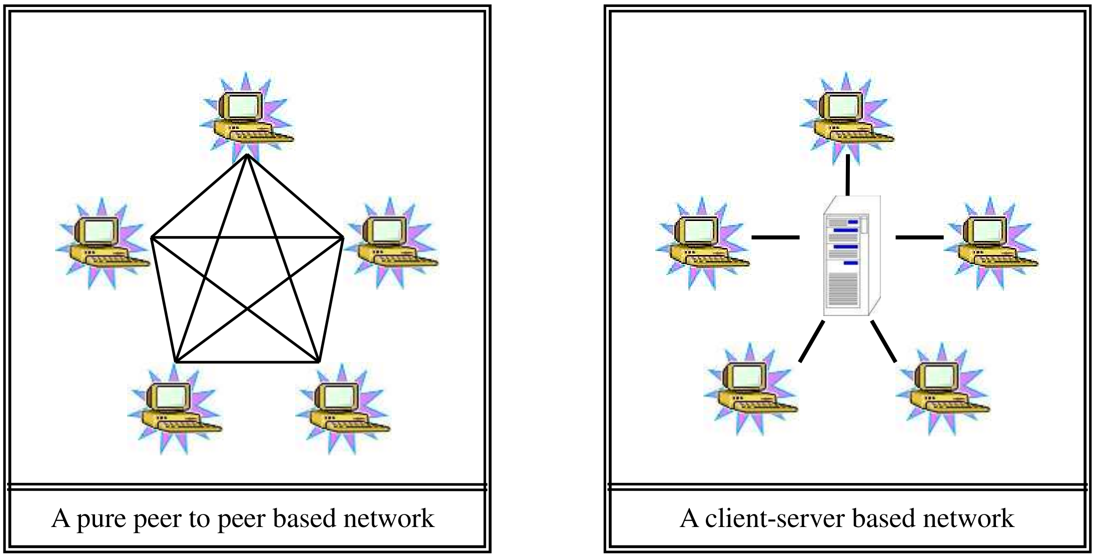
ASSIGNMENT	: Master of Science Thesis
INSTRUCTORS	: Siyu Tang
	Dr. F. A. Kuipers
SUPERVISOR	: Prof. Dr. Ir. Van Mieghem
DATE	: July 10th, 2009

Table of Content

List of Figures

1 Introduction

With the broadband surge and flourish of multimedia, peer to peer (P2P) technology becomes an important mean for mass multimedia delivery. P2P network is in contrast to traditional pure client-server model. Peers in P2P network connect each other directly for the sharing and exchanging of data. They are connected either by direct links or virtual links which depending on the routing algorithm the application using. And peers are active for both uploading content to others as a server and downloading content as a client. Various applications based on P2P technology are wide spread today, not only for file content sharing, such as audio, video, but also for real time data like telephony traffic.



A pure peer to peer based network

A client-server based network

Figure 1.1: The P2P network and client-server network.

One of the main objectives of P2P technology is making all the participants to provide resources, including bandwidth, memory and computing power. It means with the rise of the peers' number, the whole system capacity increases coinstantaneous. This is what the pure client-server system can not achieve, because in client-server system, the increase of clients means increase the load on the server linearly or lower transmit data rate for all the clients. Figure 1.1 shows these two different network systems.

Robustness is another advantage of P2P network. Peers always replicate data from multiple peers, so failure of single peer will not influence the system.

Nowadays more and more different kinds of peer-to-peer applications are in use. Some support inter-person communication like use video and text messaging, while others provide data sharing capabilities.

Typical paradigms of P2P application are in the following three fields:

- Bulk file sharing over Internet: normally achieved by unicast, meaning a direct link has been set up between peers to start downloading, e.g. BitTorrent-based software.

- Video-on-demand applications (VoD): In VoD, content is pre-recorded. User can select content and start watching the video or audio at the moment they like. User will stop watching if quality of service (QoS) not good enough.

- P2P live TV or media streaming: Applications of live media streaming are variable and popular, such as Internet television, distance education, sports events broadcasting, online games, and etc.

An early way to realize files or media streaming delivery to a group of nodes is IP multicast, especially for live media streaming. This solution is efficient in terms of resource consumption or bandwidth usage. However, it requires deployment of a large number of infrastructures or hardware, which is costly, and thus slows down the development of IP multicast.

P2P technology gives a feasible way to break the bottleneck of IP multicast to deliver the media streaming. P2P based streaming system has the advantages of requiring no especially infrastructure and being able to act as the resources of the network by increase with the number of end users. Therefore, P2P software based on live streaming developed rapidly in recent years, some software also support users to broadcast the own content.

1.1 Background

1.1.1 P2PTV

P2PTV is peer-to-peer television. The distributed video streams are typically TV channels from all over the world and some other sources. Because the distribution property of P2P technology. More end users mean better video quality of the channel. Development of P2PTV grows rapidly these years, especially in China, such like PPLive [1], SopCast [2]. And these P2PTV applications are generally deployed use two architectures: tree-based system and mesh-based system.

In tree based system, peers are constructed as a hierarchical tree. The root of the tree is the stream provider. Peers in the tree based system select their own parents to download the steam. Stream spreads continuously from the source down to the tree. This means the delay of a peer depends on the delay of route from the source to it. The positions of peers are almost static. So the delays of them are constant, and start-up time is always small. The end leaves of single tree system only receiving data, but do not upload anything to other peers. That is because they have no children peers below them. In system, the entire load is supported by interior nodes and stream source. So if a leaf in the system leaves or fails, it will not influence the system, because it has no sub-tree. But if an interior peer in a single tree system fails or leaves all other peers below its architecture will lost data until the tree structure be repaired. Even if the computational or bandwidth resources of an interior peer can not achieve the required level, his sub-tree peers will suffer from high delays or can not receive data.

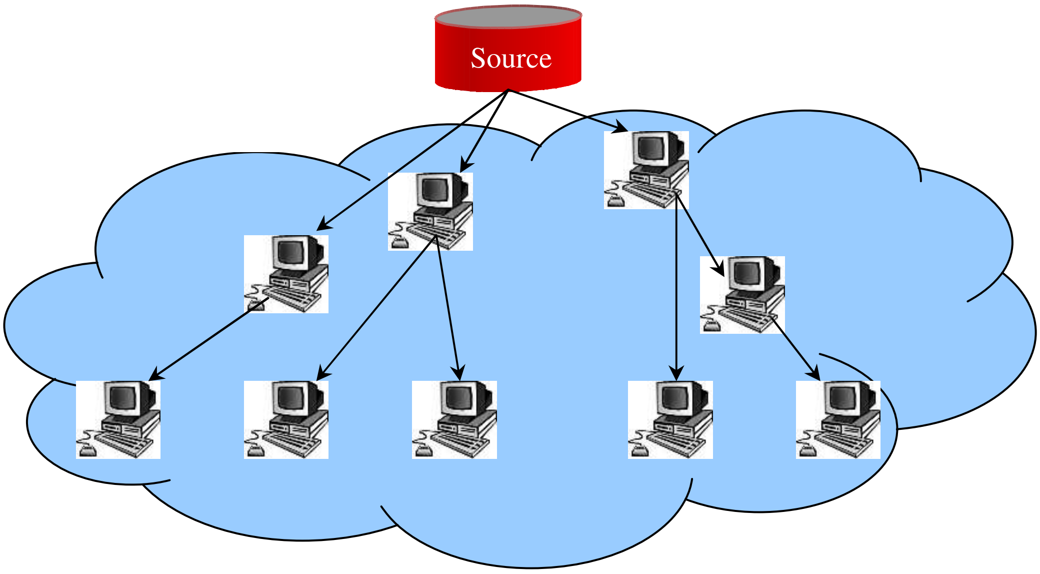


Figure 1.2: Single Tree Based Network.

In the mesh–based systems peers are not organized in a hierarchical structure like in tree based system. Peers can contact other peers they want in this network to build the neighbor lists and receiving data. The stream is forwarded as a continuous flow of data. In this system, the source splits the packets into chunks and delivered them to different peers. Peers ask for their request chunks from different potential parents, we called this mechanism as data driven. Mesh based network performs better because it is more robustness than tree based network. No single peer failure will influence the whole network here. Peers can ask data from other parents when one of them failed. So it still operation as usual without repair mechanism. So in mesh network, every peer has multiple parents and multiple children.

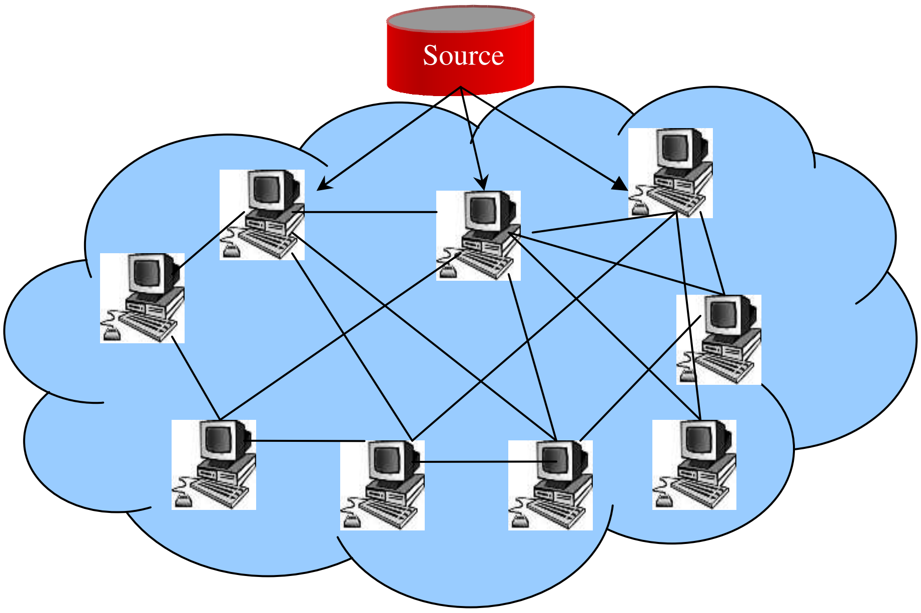


Figure 1.3: Mesh Based Network.

Many deployed P2PTV services use this architecture, such as PPLive, SopCast, CoolStreaming [3], TVAnts [4] etc. In this thesis, we focus on one of them—SopCast. Many researchers refer to SopCast as a BitTorrent P2PTV system [34] [12], but some [21] found SopCast only use the same algorithm of BitTorrent at the beginning. After that, a decentralized peer discovery process is used instead of the tracker (The definition of tracker is in Chapter 1.1.2). We will introduce the BitTorrent in the following content. Anyway, the architecture of SopCast is litter known for us now. All the conclusions are based on the experiments' results and without substantiating.

1.1.2 BitTorrent

BitTorrent [5] is one of the most popular P2P protocol based file swarm for data transferring. Figure 1.4 is the comparison of different P2P networks. BitTorrent developed by BitTorrent Inc. founded by Bram Cohen. It is a file distribution protocol which tries to divide the burden of looking up for the resources to the user.

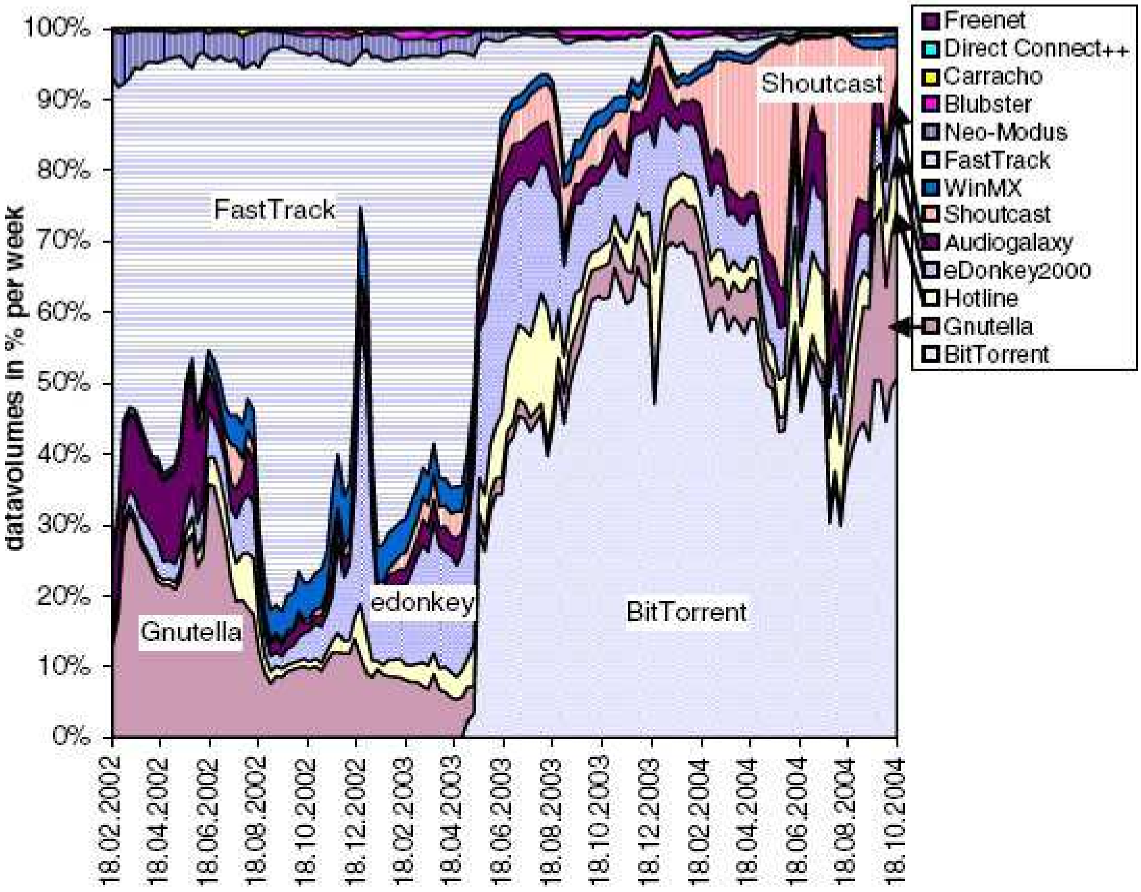


Figure 1.4: Comparison of different P2P networks.

Source: R. Steinmetz and K. Wehrle : P2P Systems and Applications, LNCS 3485, pp. 17-23, 2005.

BitTorrent Protocol is constructed over Transmission Control Protocol/Internet Protocol (TCP/IP), and transferred using Hypertext Transfer Protocol (HTTP) or HTTP Secure (HTTPS). It identifies content by Uniform Resource Locator (URL) and is designed to integrate seamlessly with the web. The advantage of it is that when multiple downloads of the same file happens concurrently, the peers upload to each other. Which make the broadcast server of the file source can support a scalable network with only a slowly increase in its load. There are three entities in BitTorrent network: a *tracker*, *metainfo* (.torrent) file and peers. .torrent file, also called seed file is a text file which included the information of tracker and file source. The principle is to divide the target file into chunks. The size of each chunk is 256 KB in BitTorrent System. A hash is computed on each chunk. These hashes are for peers to verify chunks and are added in .torrent file. Besides this, information of tracker in .torrent file includes the Uniform Resource Identifier (URI) of the tracker and the options setting.

When peers want to download a file, they have to get the .torrent file first. Peers get the URI of tracker by resolve the .torrent file. Then contact the tracker. Tracker responses the peers' request and provides the IP of other peers, includes the data source. After that, the peer will connect to the other peers, and tell each other the chunk they owned by .torrent file. Then downloading the chunks they missed from each other.

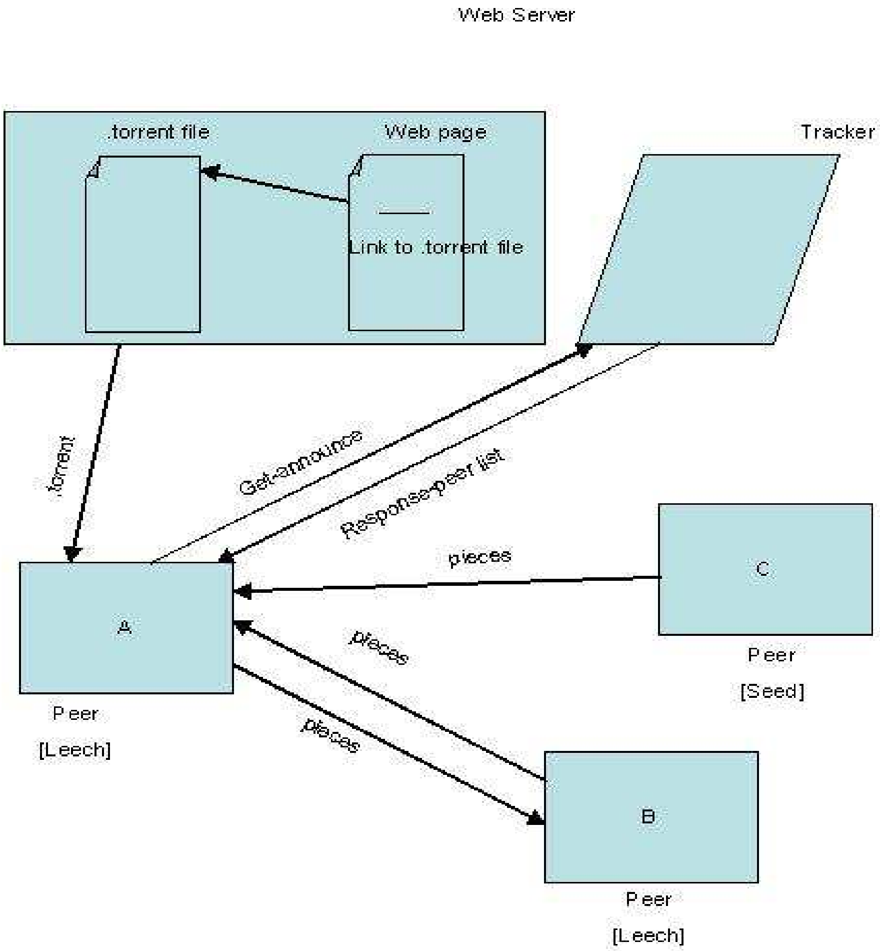


Figure 1.5: Working of the BitTorrent network.

Source: Tamilmani, Karthik (2003-10-25). "Studying and enhancing the BitTorrent protocol" (DOC). Stony Brook University. Retrieved on 2006-05-06.

Peers need to decode the hash code when they download a chunk, and compare it with .torrent file, to check whether they download the correct chunk.

In BitTorrent, peers are divided into two kinds, *seed* and *leech*. Seeds are the peers who have the complete file copy and provide it to other peers to download. Leeches are peers who have part of the file or nothing, want to download content from other peers. A leech can become seed after it got the complete file copy. The good performance of download only guaranteed if most leeches become seeds once it got the complete copy of file. But in real condition, this behavior is not common. Many peers used the network resources but leave the network immediately once they got the complete file.

The fairness of BitTorrent network is maintained by *tit-for-tat*. This means a peer can actively participate in the system only if it shares content that is of immediate interest to other peers [6]. It is implemented by "*interest/chock/unchock*" mechanism. Each peer has responsible to upgrade its download rate. So it supplies the equal level resources to peers which it downloaded from. This means only supplies services to peers who provide resource to it. But chock the peers who not cooperated. This chock strategy is temporary, which means the upload chocked, but the download can continue. The link needs not to build again when it unchocked. There is another mechanism called *optimistic unchoking*. Each BitTorrent peer has a list of preferred peers to provide resources to. These preferred peers are peers who have good link with this peer. In optimistic unchoking, a peer picks another peer which is not among the preferred peers list and upload to it. If it in return and with higher upload rate than preferred peers, it will replace the slowest preferred peer with this new one. There exits a situation that a peer may new join the network. Optimistic unchoking helps these new peers. Peers with content will allow the non share peers or new join peers download a portion of content, the receiving peer can share the same portion of data received so far to exchange with other peers. Peers who utilized the resources of other peers without contribute their own resources in exchange are called "*free rider*". [7]. Previous survey [8] shows that on the Gnutella network, nearly 70% of Gnutella users share no files, and nearly 50% of all responses are supported by the top 1% of sharing hosts. And in BitTorrent system, peers will download the rarest chunk first.

Besides the benefits, BitTorrent also has limits. For example, tracker is a vulnerable point. As known as us, the failure of a single peer will not influence the network. But the failure of tracker will make the new peer can not join the network any more.

1.1.3 Available P2PTV Applications

PPStream

PPStream [11] was founded by two engineers in Sichuan Province, was announced in the year 2005. PPStream is the first Internet video platform in the world that integrated “P2P live” and “video-on-demand” technologies. Its client software can be used as a webpage or as a desktop program. PPStream is similar to BitTorrent. PPstream provides around 90 channels. It can broadcast TV programs stably and smoothly to broadband users. In 2008, PPStream became the real Chinese largest network television platform.

PeerCast

PeerCast [23] is open source P2P radio software. It was announced in the year of 2002 in Japan. Compare to other P2P software, PeerCast downloading streams instead of files, these streams exchanged in real time with other clients. Therefore, it does not need store any data in local equipment which connected to the network.

CoolStreaming

CoolStreaming [3] is a peer-to-peer television technology which let clients share television contents to other users. The core technology is similar with BitTorrent. CoolStreaming also called DONet, it is a data driven overlay network for media streaming. The benefits of this method is can maximum the utilization of bandwidth and achieve the best performance. But this software is currently in testing period, so it is not particularly stable.

TVants

TVants [4] is also a P2P based TV network software. User of TVAnt can set up their own private tracker server database to collect different channels or share their hard disk video local files with anyone online with broadband connection In January 2007, TVants cooperate with Dream Windows to develop a powerful podcast system – Ccants Vlog. It is the first streaming video sharing system in the world, also based on P2P technology.

Zattoo

Zattoo [27] has been designed by Swiss. It is current focus on European channels. It adding fee based channels to its list. It is also designed to only allow particular users to watch particular programs. At present the service is restricted to Switzerland, France, Spain, Germany, Norway, Denmark and United Kingdom, but it expected to extend the services to other European regions. The drawback is we can not use Zattoo on Linux system.

SopCast

SopCast is simple, easy-to-use software. SopCast uses P2P technology to transmit data very efficiently. Every client can use low cost to broadcast their own program. So video broadcasting no longer need a super-server and a large amount of bandwidth. Individuals can build a good network live system compare to large-scale commercial websites. SopCast is the objective application in our project, we will introduce it more detail in Chapter 3.

Tribler

Tribler [28] is open source P2P software which design by Delft University of Technology. Tribler is also a BitTorrent protocol based application, but added multiple features mainly focused on social interaction and P2P video streaming such as recommendations, friends-aided download, and video-on-demand. So Tribler does not need another website to discover content. Tribler can use under both Linux and Windows Systems.

LiveStation

LiveStation [29] is a P2P technology based platform which developed by Skinkers Ltd. It has bandwidth limit, minimum 256Kbps for upstream and 800Kbps for downstream. LiveStation users can add some popular channels from a directory, and also can add their own option of channels to the directory.

1.2 Motivation and Objective

In P2P network, users form an overlay on the top of underlying layer. The overlay locates in the application layer of Open Systems Interconnection Reference Model (OSI model), and the underlying layer locates in the network layer. Figure 1.6 is the graph illustration of this instruction.

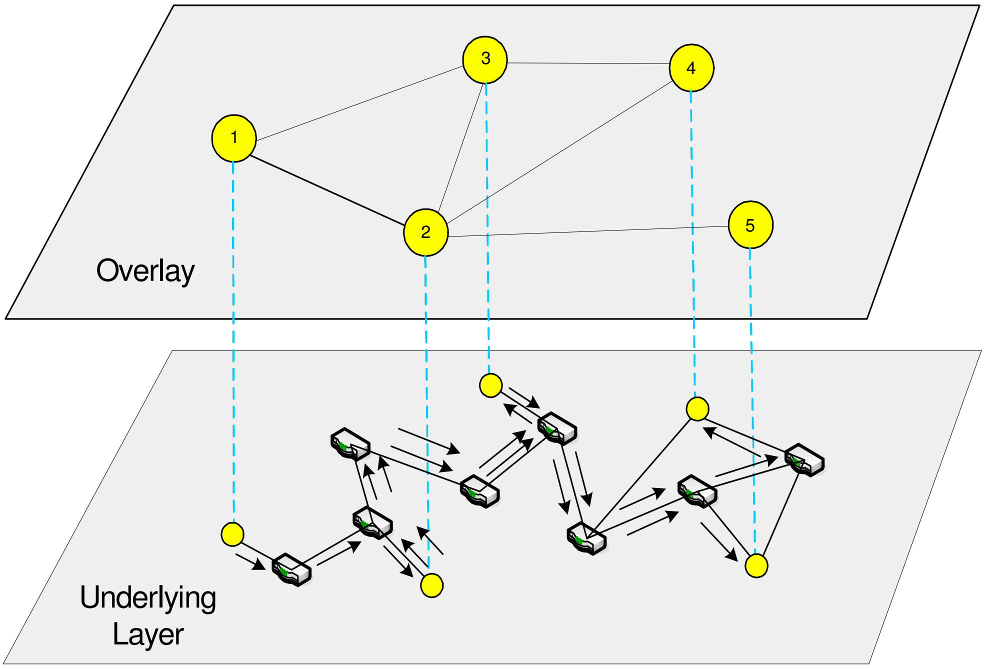


Figure 1.6: The two layer structure of P2P network.

The simple way to realize data delivery is to divert routes on the overlay without considering the underlying layer. But how the overlay constructs and how it works is a quite attractive problem. Different protocols has different incentive mechanisms and peer selection algorithms. Many researchers are working with propose the efficient algorithm to upgrade the application performance. But before that, the understanding of exists protocol for P2P network is necessary and important. Most of the P2PTV applications today are proprietary. The source code is unknown for us. So the only way to study the P2P network is through understanding the behavior of it. Many people complained the P2P network is scalable. The traffic pattern of overlay and underlying layer is unknown for us. So our objective is to understand the traffic distribution and topology of overlay, and the peer selection algorithm. Besides of this, I also try to deeply understand the dynamic overlay of P2P system, which is focus on the impact of peers' dynamic. In our project, the P2P system which we selected is SopCast. I will introduce it in Chapter 3.

1.3 Thesis outline

The rest of this thesis is organized as following: Chapter 2 is the literature survey, in which I will introduce the measurement methods and results which other paper finished. In Chapter 3, I describe SopCast and PlanetLab [9] which we used for our experiment set up and how we conduct our experiments. Chapter 4 presents the analyses of traffic pattern and peer selection in a stable network. And Chapter 5 is the analyses of the impact of dynamic overlay of the SopCast system. The last part is the conclusion of our results and the future work which we can deploy.

2

Literature Survey

2.1 P2P Network Measurement and Development

Sliverston and Fourmaux [10] observed the traffic patterns the P2PTV generate and the underlying mechanism they used to understand the behavior of P2P IPTV systems. The measurement performed during the 2006 FIFA World Cup. That is because it is a large-scale event, and they can collected data from large-scale P2P networks. They chose the most popular applications on the internet as the objectives: PPLive, PPStream [11], SopCast and TVAnts. From their study, they show the traffic patterns and mechanisms to get the video are different for different P2PTV applications. They also maintain different peers' neighborhood and peers use different download policy to get the video. They also analyzed the video peers' lifetime for all applications.

Fallica et al [12] set a network with 70 nodes on PlanetLab, measured the functionalities and the characteristics of SopCast and the Quality of Experience (QoE) with objective and subjective measurement technologies. The results characterized the behavior of SopCast. They also concluded the QoE part, which is that SopCast can provide good quality video to peers broadcasting from a PC but suffers from peer lags. The zapping time in SopCast is significantly high.

Zhang et al [3] proposed a data-driven overlay network, which is called DONet. DONet was implemented yet, which called CoolStreaming. It was released on May 30, 2004. CoolStreaming's core operations are the same as in BitTorrent. In this paper, they presented the design of DONet, proposed an efficient member and partnership management algorithm, together with a scheduling algorithm for real-time and continuous distribution of streaming content. They also point out the pratival challenges for realizing DONet. And extensively, they evaluated the performance of DONet over PlanetLab. It is already attractive more than 30000 distinct users.

The statistics and feedback from users are positive.

Ali et al [13] evaluated the performance of two popular proprietary video streaming systems: PPLive and SopCast. They set up experiments to look insight the operation of these systems from a small number of vantage points. They collected the data from runs of the system under different environment. And separate the packets into data packets and control packets when they analyzed data they collected. Their result shows the greedy algorithms they used and the locality independence of the underlying model leads to very high resource usage. They also found these applications have not implemented the tit-for-tat fairness, and the data distribution structure is built randomly without considering the bandwidth.

2.2 Peer Selection Policy

Shah et al. [14] proposed a P2P solution for multimedia streaming based on BitTorrent protocol. As we known, BitTorrent is a content-distributed protocol. The modification if because the BitTorrent does not consider the real-time stricture the streaming application needed. In BitTorrent, peers do not download chunks in sequence, this make the chunks can not use until the download completed. And tit-for-tat policy forces too many peers to wait for too long before download chunks. For these two weaknesses, they proposed the modifications: chunk selection policy and neighbor selection policy. First, they replaced the policy that rarest chunk downloading first of BitTorrent by a policy that preventing peers from requesting chunks they will not watch in the near future. Second, they introduce a new randomized tit-for-tat peer selection policy that gives free tries to a larger number of peers and lets them participate sooner in the media distribution. Their simulations results indicate that both modifications significantly improve real time problem in multimedia streaming.

Horvath et al. [15] did the experimental to characterize the traffic generated by the P2PTV Application. They also study insight the mechanism that governs the data transfer. The results show that for all application, the selection of the peers to download video content is strongly affected by the peer upload bandwidth. And only in TVAnts and partly in PPLive, peer location information is exploited by the application. PPLive peers are keeping contact new peers at a fixed rate. While peers in the TVAnt and the SopCast network are slow down after the initial peer bootstrap.

Wu et al. [25] designed an optimal peer selection algorithm for minimize the end-to-end latency.

They first formulate the problem as a linear optimization problem, and then solved it by proposing an efficient algorithm. This algorithm is based on Lagrangian relaxation technique and the subgradient algorithm. And it computes the optimal streaming rates on the peer-to-peer links in a fully decentralized and iterative fashion. This algorithm also reactive to the network dynamics, such as peers' joining and failure. They also compare the performance of this optimal algorithm. The result shows this optimal peer selection algorithm has better performance in dense network, which means minimized more end-to-end latency to optimized bandwidth utilization.

Many researchers working on proposing more efficiency peer selection algorithm for P2P network. Li et al. [26] also proposed an Underlying Topology-aware Peer Selection Algorithm (UTAPS) instead of the current randomly peer selection algorithm. This algorithm implemented the idea the idea that select peers on the overlay in the range of small hopcount and low round trip time (RTT). The main objectives of this proposal are trying to reduce the cross-ISP (Internet service provider) traffic generated and upgrade the individual performance. This algorithm was implemented into two steps: Inferring the underlying topology (RTT and hopcount) and peer selection based on the inferring underlying topology. The topology inferring is finished by traceroute. After compare the result of UTAPS with the original randomly peer selection algorithm, they proved the UARTS achieve better individual performance in download time and efficient use of resources in ISP's backbone.

The measurements of P2P network until now are from several vantage points, which means does not collect and analysis the data of a complete scalable P2P network [10] [13]. And most measurement only working for the traffic pattern in a relative static environment [10] [2] [13], less do the measurement under the dynamic condition. Our project not only does the measurement under the static environment, but also working under the dynamic environment. We look inside the structure of the SopCast network, and try to understand how this system works.

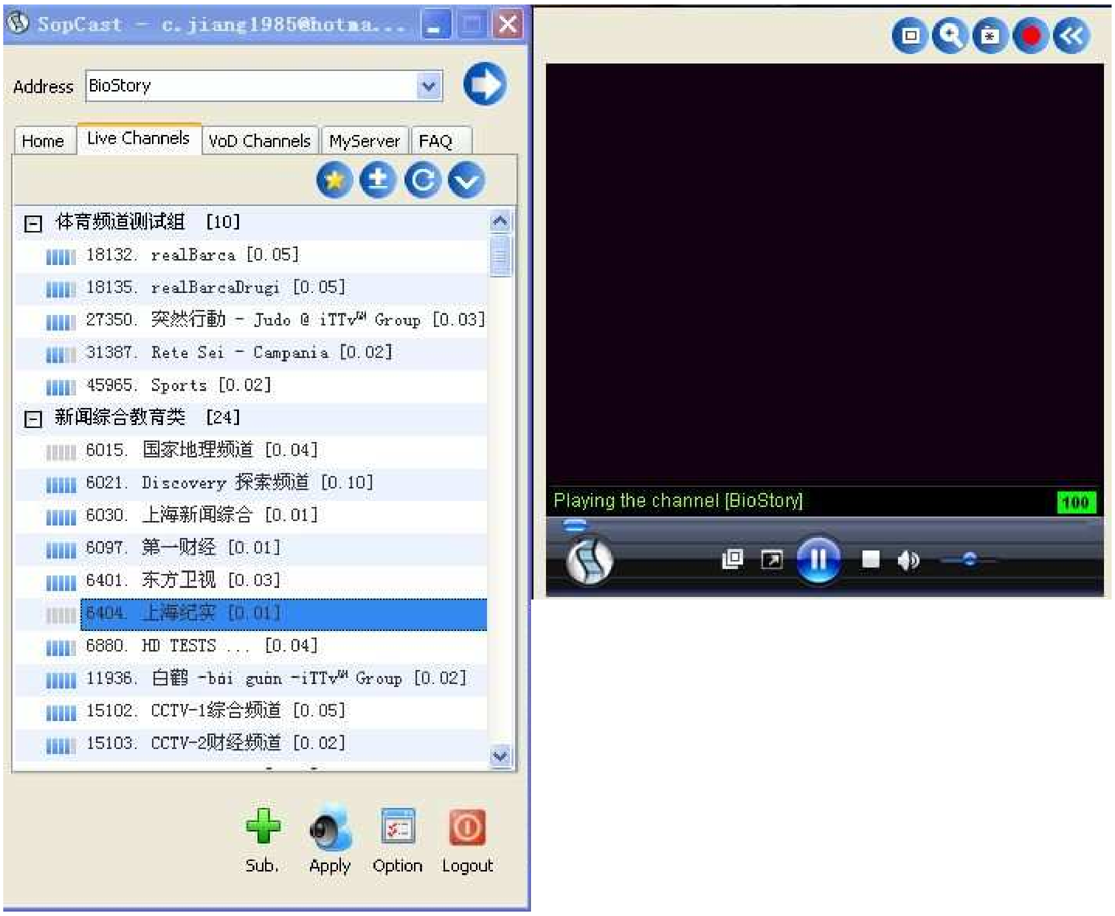
3

Experimental setting on PlanetLab

3.1 The SopCast Streaming System

In this project, we study SopCast, a typical proprietary P2PTV system. SopCast is a Streaming Direct Broadcasting System based on P2P technology. And is one of the most popular P2PTV applications. In Figure 3.1, we present the user interface of the client of the SopCast application and a channel list which can be access by everyone. It was developed by a student group of Fudan University in China. Users can use it to watch TV via the Internet, or broadcast their own audio/video. This means that a traditional server-client mode for video broadcasting is not needed, and excessive bandwidth consumption from the server-client mode can be saved. By using Sop Cast, an end user is allowed to build his/her own live broadcasting system. Approximately, a personal PC can support 10,000 peers concurrently by the SopCast system [2]. *SoP* is the abbreviation of "streaming over P2P". Since SopCast is a proprietary system, there is no source code available. Therefore, it is important to understand the network dynamics and traffic distribution of the SopCast network. Currently, all the SopCast services are free. The video in the SopCast system are encoded in two video formats: Windows Media Video (WMV) or Real Video (RMVB). In order to run SopCast successful, you should have at least 512K bps broad bandwidth. And 1M bps internet connection is recommended [2].

The video quality of the channels depends on how many users are watching. The architecture of many P2PTV networks can be considered as real-time versions of BitTorrent: if a user wants to watch a specific channel, the P2PTV software contacts the tracker server for that channel in order to get the addresses of peers who distribute that video content. Then, it will contact these peers to download video content. The tracker records the user's address, so that it can be given to other users who wish to view the same channel. And this forms an overlay on top of the regular internet.



⊟ 体育频道测试組 [10]
- 18132. realBarca [0.05]
- 18135. realBarcaDrugi [0.05]
- 27350. 突然行動 - Judo @ iTTv™ Group [0.03]
- 31387. Rete Sei - Campania [0.02]
- 45965. Sports [0.02]

⊟ 新闻综合教育类 [24]
- 6015. 国家地理频道 [0.04]
- 6021. Discovery 探索频道 [0.10]
- 6030. 上海新闻综合 [0.01]
- 6097. 第一财经 [0.01]
- 6401. 东方卫视 [0.03]
- 6404. 上海纪实 [0.01]
- 6880. HD TESTS ... [0.04]
- 11936. 白鹤 -bái guòn -iTTv™ Group [0.02]
- 15102. CCTV-1综合频道 [0.05]
- 15103. CCTV-2财经频道 [0.02]

Figure 3.1: The SopCast user interface.

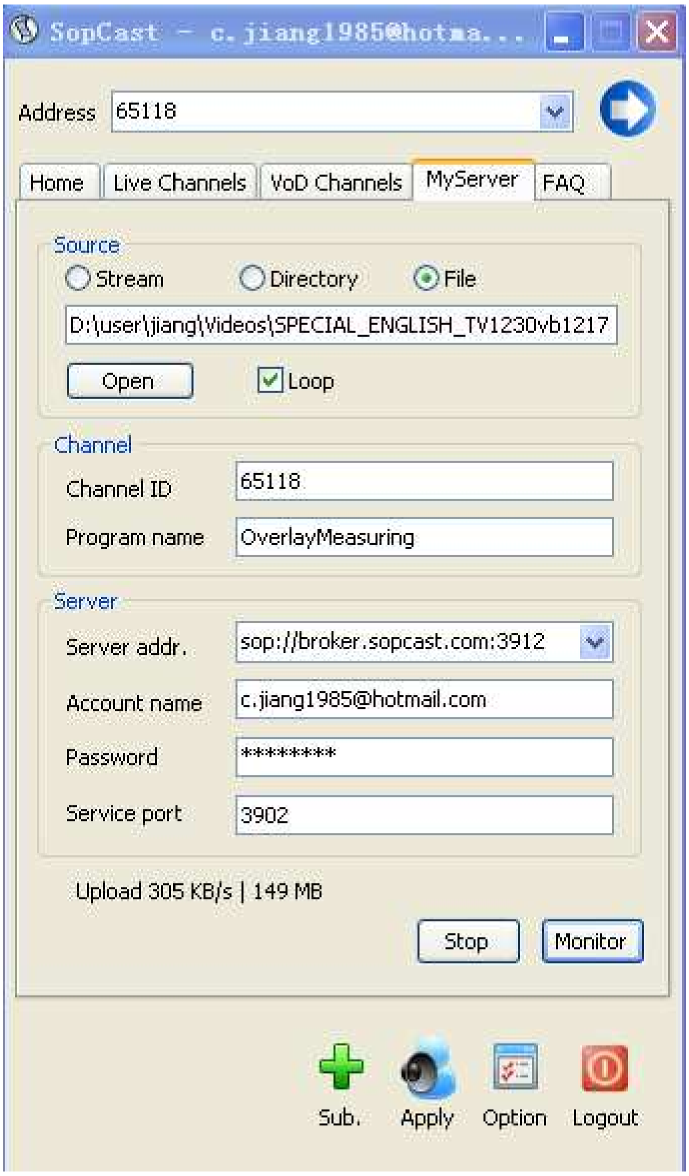
As we mentioned above, the SopCast application has the function to allow users to broadcast their own channel. "*MyServer"* in the user interface (Figure 3.1) is designed for this mission. Figure 3.2 is the broadcast interface for users. Once a user starts a channel with SopCast, it becomes the source provider and starts to build the P2P network. Before a user can broadcast our own content, he/she need to register a dedicated channel. Users can do this by click the link "MySop" on the SopCast website. In this system, you can register as a new user, upload the user data, register channel, manage channel and etc. Users can broadcast their own data once they register a channel. In Figure 3.2, we present the SopCast broadcast interface. User need to fill the parameters following:

Channel ID: User can get the channel ID after they register. The Channel Name is optional.

The Server Address: The server IP address where the channel registered.

Account Name and *Password:* Broadcaster's own account name and password to log in the SopCast system.

Service Port: The default service port is 3902. User can change it if they need.



SopCast - c.jiang1985@hotma...

Address 65118

Source

○ Stream ○ Directory ● File

D:\user\jiang\Videos\SPECIAL_ENGLISH_TV1230vb1217

☑ Loop

Channel

Channel ID 65118

Program name OverlayMeasuring

Server

Server addr. sop://broker.sopcast.com:3912

Account name c.jiang1985@hotmail.com

Password ********

Service port 3902

Upload 305 KB/s | 149 MB

Figure 3.2: The SopCast broadcast interfere.

There are three ways for a SopCast user to broadcast his/her stream source: live stream, media directory or single file [2]. Live stream feed the channel with a live stream source generated by media server or media encoder. If media directory is selected, SopCast will search all supported media files in the directory or subdirectory, and generate *sop_playlist.spl* file automatically. The SopCast will use this spl file to broadcast all these media files. Users can modify it to meet your requirement. Single file feed the channel with media file on local disk. Now SopCast supports 4 file types, *.asf or *.wmv, *.rm or *.rmvb, mp3, *.spl. In our experiment, we broadcast the single file. It is the simplest method for us to implement. And it is also a common method for normal users. The broadcaster can monitor the quality of the channel by two parameters: quality of source (QS) and quality of client network (QC). The Video on Demand (VoD) of SopCast has not implemented yet.

3.2 Overview of PlanetLab

We have used PlanetLab in our experiments, because it allows us to evaluate the entire constructed overlay under a fully controlled environment. The PlanetLab nodes run the Linux operation system, and SopCast is the only mesh based P2P applications that is supported under the Linux system.

PlanetLab is an experimental network for new network services development, especially for distributed storage, network mapping, peer-to-peer systems, distributed hash tables, and query processing. It was established in 2002 and currently consists of 1011 nodes at 475 sites. Most of peers located in the research institutes over the world, such as the top universities, lab centers. All of the machines are connected to the Internet. PlanetLab is an overlay testbed. It allows users to deploying, evaluating and accessing planetary-scale network services. PlanetLab services and applications run in a *slice* [32] of the platform: a set of nodes on which the service receives a fraction of each node's resources, in the form of a virtual machine. Users use *Secure Shell* (SSH)[1] to log on the PlanetLab nodes. The benefits of using PlanetLab are researchers can deploy new services in the real world network, and at a scare network size.

All PlanetLab nodes run a common software package. This package includes Linux based system and some other tools such as management tools that monitor node health and control system parameters. We call this packet as "*MyPLC*". The objective of the software is to support distributed virtualization [32], which make each service runs in an isolated slice of PlanetLab's global resources. This allows applications to run across all or some of the machines distributed over the globe. And in some times, as request, may running multiple

[1] SSH is a network protocol that allows data to be exchanged using a secure channel between two networked devices.

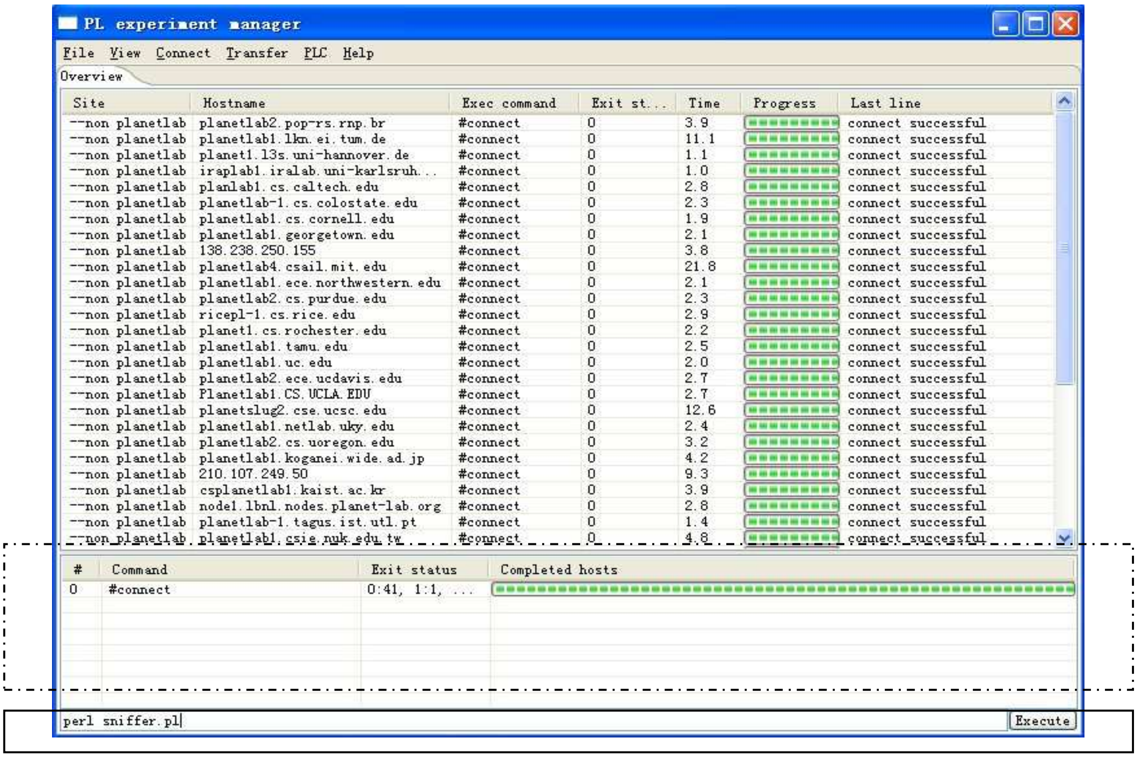
applications in different slices of PlanetLab.

The initial slices of PlanetLab are just a collection of machines connected with Internet. Users have to install their own applications they need for experiment on each peer. This process is quite trivial. Fortunately, there are several slice manage tools developed for user to deploy, monitor, and run applications on PlanetLab. They give us the ability to centrally manage, install, upgrade, start, stop, and monitor of applications on a PlanetLab slice. They make PlanetLab easier to use. Users can find these tools easily on PlanetLab website. Such as pShell [30] from McGill University and Pssh [31] developed by Intel Research at Berkeley.

3.3 Experiment Set-up

We have used PlanetLab Experiment Manager [16] (PlMan) to manipulate the PlanetLab nodes (PL nodes) in our experiment. PlMan developed by University of Washington. The PlMan has a graphical user interface (GUI) as shown in Figure 3.3. With PlMan, we can perform common tasks, e.g. log into a series of PL nodes, or upload and download file from them, etc. To run the SopCast application, or monitor the SopCast traffic on PL nodes, we have created Perl [17] scripts that can be executed in PlMan (Figure 3.3).

We can see in Figure 3.3, the peers are already connected with PlMan. We can put our command which we want to run on each peer in the solid line area (like command "*perl sniffer.pl*" in this figure). The content in dotted line area is the history list of the executed command before. With this tool, we not only run the command but also can upload files to each peer or download data concurrently. We also can monitor whether the command run successfully on each peer by check the status. But this tool also has weakness. It always has authority problem which causes its unavailability. The reason of this is unknown for us.



Site	Hostname	Exec command	Exit st...	Time	Progress	Last line
--non planetlab	planetlab2.pop-rs.rnp.br	#connect	0	3.9	🟩	connect successful
--non planetlab	planetlab1.lkn.ei.tum.de	#connect	0	11.1	🟩	connect successful
--non planetlab	planet1.l3s.uni-hannover.de	#connect	0	1.1	🟩	connect successful
--non planetlab	iraplab1.iralab.uni-karlsruh...	#connect	0	1.0	🟩	connect successful
--non planetlab	planlab1.cs.caltech.edu	#connect	0	2.8	🟩	connect successful
--non planetlab	planetlab-1.cs.colostate.edu	#connect	0	2.3	🟩	connect successful
--non planetlab	planetlab1.cs.cornell.edu	#connect	0	1.9	🟩	connect successful
--non planetlab	planetlab1.georgetown.edu	#connect	0	2.1	🟩	connect successful
--non planetlab	138.238.250.155	#connect	0	3.8	🟩	connect successful
--non planetlab	planetlab4.csail.mit.edu	#connect	0	21.8	🟩	connect successful
--non planetlab	planetlab1.ece.northwestern.edu	#connect	0	2.1	🟩	connect successful
--non planetlab	planetlab2.cs.purdue.edu	#connect	0	2.3	🟩	connect successful
--non planetlab	ricepl-1.cs.rice.edu	#connect	0	2.9	🟩	connect successful
--non planetlab	planet1.cs.rochester.edu	#connect	0	2.2	🟩	connect successful
--non planetlab	planetlab1.tamu.edu	#connect	0	2.5	🟩	connect successful
--non planetlab	planetlab1.uc.edu	#connect	0	2.0	🟩	connect successful
--non planetlab	planetlab2.ece.ucdavis.edu	#connect	0	2.7	🟩	connect successful
--non planetlab	Planetlab1.CS.UCLA.EDU	#connect	0	2.7	🟩	connect successful
--non planetlab	planetslug2.cse.ucsc.edu	#connect	0	12.6	🟩	connect successful
--non planetlab	planetlab1.netlab.uky.edu	#connect	0	2.4	🟩	connect successful
--non planetlab	planetlab2.cs.uoregon.edu	#connect	0	3.2	🟩	connect successful
--non planetlab	planetlab1.koganei.wide.ad.jp	#connect	0	4.2	🟩	connect successful
--non planetlab	210.107.249.50	#connect	0	9.3	🟩	connect successful
--non planetlab	csplanetlab1.kaist.ac.kr	#connect	0	3.9	🟩	connect successful
--non planetlab	node1.lbnl.nodes.planet-lab.org	#connect	0	2.8	🟩	connect successful
--non planetlab	planetlab-1.tagus.ist.utl.pt	#connect	0	1.4	🟩	connect successful
--non planetlab	planetlab1.csie.nuk.edu.tw	#connect	0	4.8	🟩	connect successful

#	Command	Exit status	Completed hosts
0	#connect	0:41, 1:1, ...	🟩

```
perl sniffer.pl
```

Figure 3.3: The GUI of PlMan (windows version).

Our experiments consist of two types of nodes:

Source Provider[2] (SP): SP is a standard personal computer located in our campus network. We registered a dedicated channel to the SopCast network before. In this channel, a small movie with duration of 25 minutes and size of 59.1 Mbytes is continuously broadcast in a loop. The SP runs Windows XP. It is equipped a 10/100 FastEthernet network interface. In order to determine the route taken by packets across the IP network from SP to other peers in this network, we run the tracert [24] program on SP. The tracert program runs continuously uninterruptedly during the experiments. The traffic collection at the SP is accomplished with Windump [18].

Peer: Peers are the PlanetLab nodes who are viewing the TV channel released by SP. These nodes should be active in downloading and uploading process. In this thesis, the term of peer and node are interchangeable. Each of the 50 nodes under consideration runs the following software: (1) SopCast Client (Linux version), with command line control; (2) Tcpdump [19], a packet sniffer to passive monitor the traffic transmitted at the SopCast peers;(3) traceroute [20], traceroute is the Linux version of tracert. With the same objective of running tracert on

[2] Source Provider is the node who broadcast the entire video by using SopCast software.

SP, we run traceroute on each peer to determine the route taken by packets across the IP network among peers. The traceroute program also runs continuously uninterruptedly; (4) Perl scripts for the manager to remotely control the PlanetLab nodes behaviors.

We designed two scenarios for our objectives. In the first scenario, we study the SopCast network in a relatively stable environment, meaning, nodes joining the SopCast network simultaneous without departure till the end of the experiment. In the second scenario, we aim to evaluate how the dynamics of the SopCast system may affect its traffic pattern.

4

The Video Delivery Pattern and Peer Selection Criterion in the SopCast Network

4.1 Scenario Description

In the first scenario (*scenario I*), we want to understand the traffic distribution as well as the peer selection algorithm in the SopCast system.

In Scenario I, we conducted a single experiment lasting roughly 45 minutes. The 50 peers were controlled in such a way that they joined and left the network simultaneously. We collected the traffic log files captured by tcpdump/windump and by traceroute/tracert from all the 50 peers and SP. The collected files are further processed and analyzed with AWK[3] script. In the first part of scenario I, the bandwidth limit of PlanetLab nodes are all not pre-defined. But for researching the impact of bandwidth limit, we limit the bandwidth of each PlanetLab nodes in part 2 of scenario I.

4.2 Stability of the Paths in the Underlying Network

The SopCast P2PTV system is a highly dynamic network on the overlay layer, with frequent link establishment and termination. To start with, we examine the impact of the SopCast traffic on the underlying network.

The output file of traceroute contains the information of each router along the path and the round-trip time (RTT). RTT is the time required for a specific source sending a probing

[3] Awk is a programming language that is designed for processing text-based data, either in files or data streams.

packet to a destination until getting the Internet Control Message Protocol (ICMP) reply packet. It includes the forward time and backward time.

We define a *dominant path* as the path transferred the largest amount of packets between a node pair and the *temporary path* as the other paths between this node pair. We run the traceroute continuously without interrupt during the experiment period to collect the information of all dominant and temporary paths and RTT on the underlying layer. From the files captured by traceroute, we found that the path between a node pair is quite stable. Around 98% node pairs transfer their data packet only using one path, i.e. the dominant path.

4.3 The SopCast Traffic Distribution

The SopCast overlay is constructed with decentralized peers. As we introduced in Chapter 1, SopCast is a chunk based system. TV content is divided into video chunks or blocks with equal size. If a peer is providing blocks to other peer, we call it *parent*. If a peer receiving video blocks from another peer, we define it as *children*. Each peer has a number of parents from which it can download video and a number of children to which it should provide the video blocks. And the traffic throughput involved in the analyses of Chapter 4 and Chapter 5 only aims to the video packets delivery, which not includes the control packets traffic.

Previous studies [21] on SopCast have shown that some nodes has larger upload throughput than download throughput and these nodes are all connected with SP directly. In our experiment, we have noticed similar observation, which is presented in Figure 4.1.

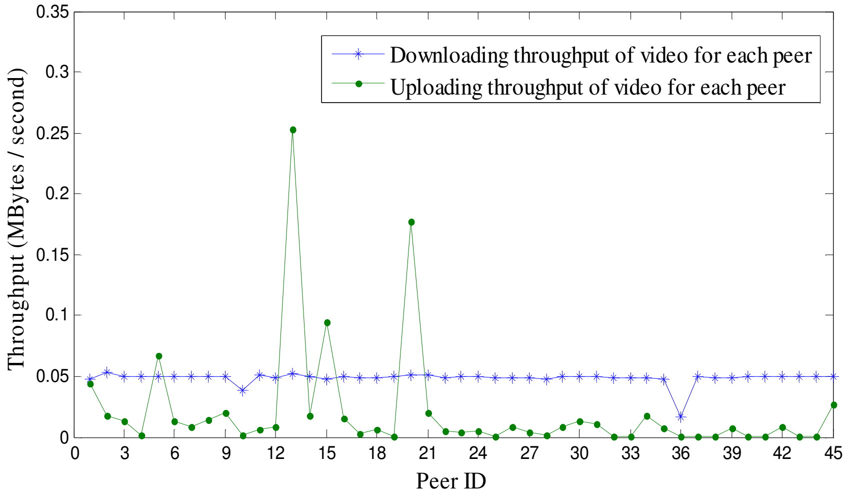


Figure 4.1: The download throughput and upload throughput for each peer.

In Figure 4.1, we can see that the download throughput at each peer is almost uniformly distributed, while peers behave differently regarding the uploading throughput. It seems there is a lack of tit-for-tat fairness in SopCast. As we introduced in Chapter 1.1.2, if tit-for-tat is implemented in this application, peers which has less or zero upload should not have so much download video. Which means the upload throughput should be even if the download throughput is a uniform distribution. But from Figure 4.1, we found some peers almost has no upload content, but also has a good download throughput to support the service.

In contrary to these peers, we also found there are a small number of peers upload much more than what they have downloaded. And these peers connect to the SP directly. We have also notice that there are several peers download from the SP, while they do not upload a lot to the network. The remaining peers, which have never connected with the SP during our experiment, generally upload a few video contents. We define *super-peers* as the peers connected with SP directly. Based on these observations we can visualize the SopCast network as a three-layer structure.

From the above analysis, we can visualize the video transmission in the SopCast network as a three layer structure, as shown in Figure 4.2.

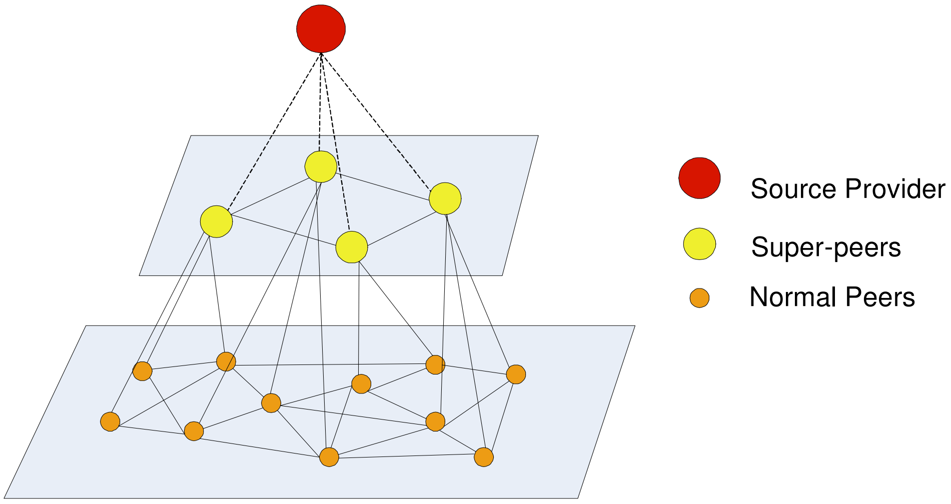


Figure 4.2: The three layer structure of SopCast network.

In Figure 4.2, the SP owns the entire video content in the network. It only provides the video stream to a limited number of peers, that is, the super-peers. There is no video transmission from the super-peers to the SP. The super-peers act as the major force to provide video blocks to other peers. The burden of the SP is alleviated by them. However not all of them contribute the video content blocks to other peers as a sub sources, only part of them has larger upload throughput than download throughput. The remains have larger upload than normal peers, but still lower than their download throughput. Normal peers are peers which have not connected with SP during the period of experiment. This three-layer structure only regards the video transmission.

As the importance of super-peers in the SopCast P2P network, we want to discuss the traffic distribution of them specially. In Figure 4.3, we plot the traffic distribution at the super-peers in more detail. We found the download throughputs of super-peers are mainly provided by SP, and a part exchanged among the super-peers. The video stream from normal peers to super-peers is quite a small part. This means super-peers seldom download data from normal peers. They download video content from the SP and other super-peers. So the links between super-peers and normal peers, links among super-peers and links among normal peers are all bidirectional (solid line in Figure 4.2).

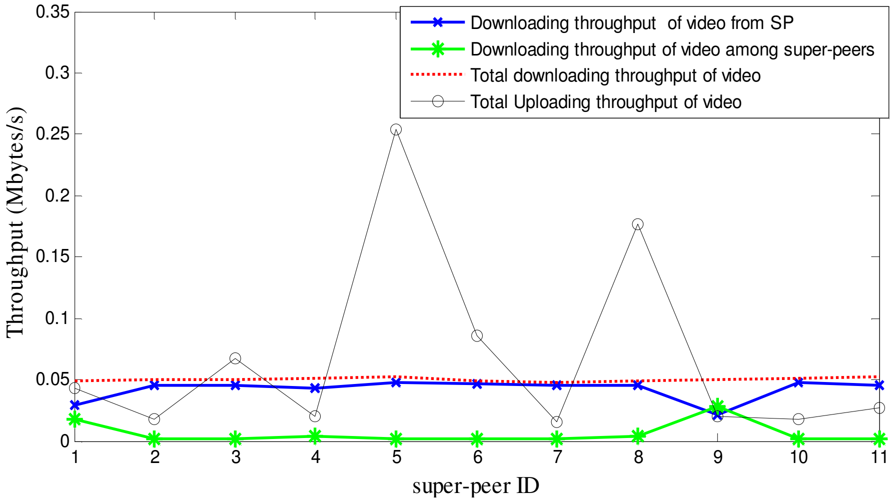


Figure 4.3: The detail of super-peers' traffic distribution.

4.4 The Super-peer Selection Algorithm in SopCast

From the analysis above, we can see that the super-peers play an important role in the SopCast system. They share the loads of SP to provide video content to other peers. In the sequel, we aim to answer the question, that is, how does SopCast chooses the super-peers?

Many designed peer selection algorithms are based on the underlying topology [25] [26] or peer heterogeneity such as capacities. It makes sense to choose these parameters as peer selection factors. Because these kinds of algorithms always can save time consumption and resources consumption, or can allocate the traffic load to each peer fairly, then to utilize the resources better. So we proposed the super-peer selection algorithm may also designed by one of these mechanism.

We selected the possible factors which may impact the super-peer selection.

- Delay/hopcount
- Capacity/bandwidth
- Registration process

4.4.1 The impact of the delay and hopcount

First of all, we examine the delay and hopcount. In this analysis, we calculated the average value of RTT as delay metric. In Figure 4.4, we show the hopcount distribution between the SP and each peer, and the hopcount distribution between the SP and super-peers. Figure 4.5 is the average RTT from SP to each peer. We marked the super-peers in this figure. If the hopcount or delay impacts the super-peers selection, the SP should select the peers with *shortest path*, or peers with smallest delay. In this scenario, we have assumed the shortest path as the path with minimal number hopcount. But Figure 4.4 and Figure 4.5 clearly show that the SP has not selected the super-peers with these two metrics. We can see hopcount distribution has no discipline. In Figure 4.5, we ordered the delay which between SP and each peer. We can see the super-peers selection shows totally no rule in this figure. This means super-peers selection algorithm did not based on the hopcount and delay between SP and peers.

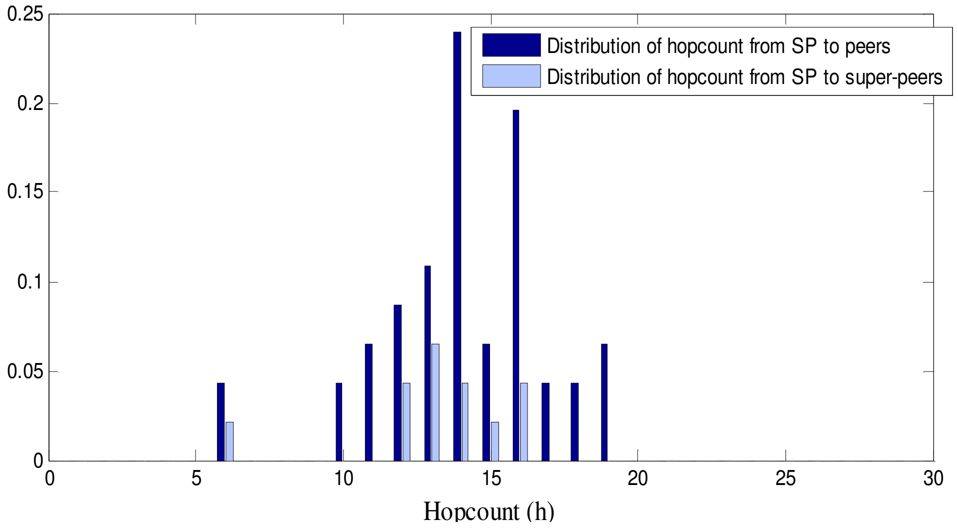


Figure 4.4: Hopcount distribution from SP to each peer and from SP to super-peers.

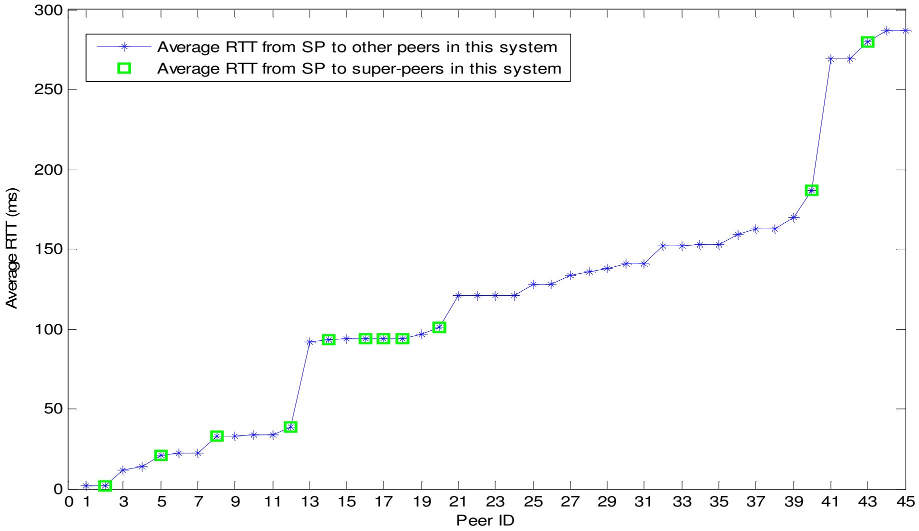
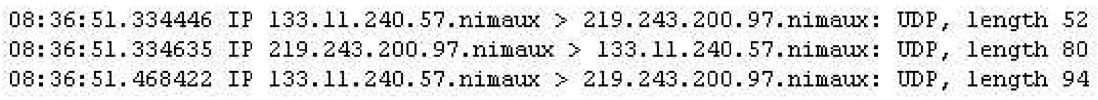


Figure 4.5: Average RTT from SP to each peer, super-peers marked with square.

4.4.2 The Impact of the Registration Process

Registration process is another factor which may influence the super-peer selection algorithm. We show times of registration process in Figure 4.7. The registration time here is the value that peers first send the 52-byte contact packet to the SP. In principle, peers of PlanetLab should synchronize their time to Eastern Standard Time (EST) [9]. But many peers still has significant shift from the accurate EST time. That is because PlanetLab using Network Time Protocol[4] (NTP) to synchronize the time. But NTP causes influence with the system clock on several peers. CoMon [22], which is a monitoring infrastructure for PlanetLab, collected the time shift for each peer. The peers involved in our experiment are within 10 second range shift. For the preciseness of results, we collected the values of time from log file of SP to present the registration time at the SP. Broker server is the tracker of this SopCast system. It is located in Shanghai, China. The introduction of tracker is in Chapter 1. Because we can not access broker server, we use the time displayed at the log file at each peer to present the registration time at broker server. Log files are captured by tcpdump/windump. The result displayed as the format in Figure 4.6.

[4] NTP is a protocol designed to synchronize the clocks of computers over a network.



```
08:36:51.334446 IP 133.11.240.57.nimaux > 219.243.200.97.nimaux: UDP, length 52
08:36:51.334635 IP 219.243.200.97.nimaux > 133.11.240.57.nimaux: UDP, length 80
08:36:51.468422 IP 133.11.240.57.nimaux > 219.243.200.97.nimaux: UDP, length 94
```

Figure 4.6: The result format of Tcpdump/Windump.

The first column is the formatted time. And the second column is the network layer protocol. The following are the source and the destination of transmitted packets. Column shows "UDP[5]" is the transport layer protocol. And the last column is the length of the packets.

In order to make the following analyses easier, we unformatted the time, and calculated the relative time of each peer. Relative time is the seconds from the first capture. So the time value in this figure is the relative time.

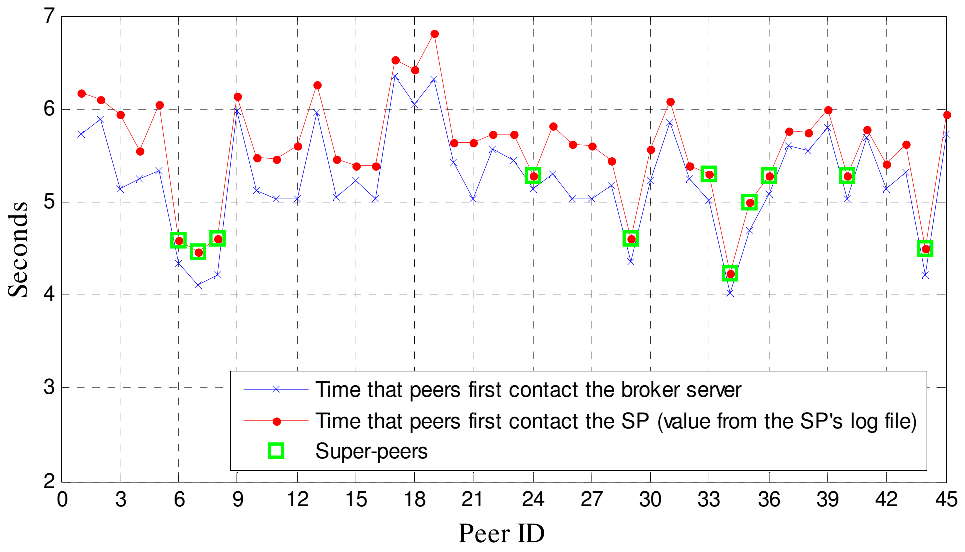


Figure 4.7: Registration time on SP and broker server.

We marked the super-peers in Figure 4.7. After check the super-peers distributed, we can found that super-peers selection is strictly depends on the time that peers first contacted the SP. This means the super-peers selection is indeed affected by the registration time of the SopCast nodes at the SP. We refer the super-peer selection algorithm which relies on the registration times on SP as the *registration time-based algorithm.*

[5] UDP: User Datagram Protocol.

4.4.3 The Impact of Bandwidth

To verify whether the bandwidth impact the super-peer selection, this experiment only includes the peers with pre-defined bandwidth limit (always defined by the administrator of nodes.). We have 44 peers with 10 mbit/s and 6 peers with 1gbit/s. The bandwidth limit of peers is collect from the XML[6] file. It contains all the configure information of each peer, such as URL, IP, MAC address[7], bandwidth limit. The bandwidth limit here is the resource for all the slices on this node. In order to calculate the available bandwidth, we collected the consumptions of other slices on these nodes from CoMon. It has the record of TX rate and RX rate of each peer in the previous three days. And it updates per 5 minutes. We collected the TX rate and RX rate values of the previously 5 minutes before experiment. The remaining bandwidth for each peer is calculated by using the total bandwidth minus the sum of TX rate and RX rate consumed by the other slices. The nodes we selected are all only consumed less than 3kbit/s bandwidth. It means the deviation is from 0.0003% to 0.03%, so we ignore this deviation in our scenarios. The result of time and super-peers are in Figure 4.8.

During the experiment, there are 10 nodes stopped to being active in the network. The reason is unknown. These peers include 1 peer with 1gbit/s bandwidth limit and 9 peers with 10mbit/s bandwidth limit, we excluded these peers in this figure. In Figure 4.8, we present the registration time of the peers in our experiment. The peers are ordered with increasing registration time on the x-axis. We found in this experiment, there is two peers reply the packet "*service port unreachable*" to SP. This means the router does not have a route entry for incoming packets. Hosts in PlanetLab are time synchronized via NTP, so when this happens, the router will send an ICMP network unreachable error message back to the source to indicate that the destination is unreachable if it is allowed to do so [33].

[6] XML (Extensible Markup Language) is a general-purpose specification for creating custom markup languages. *http://www.planet-lab.org/xml/sites.xml.*

[7] Media Access Control address (MAC address) is a unique identifier assigned to most network adapters or network interface cards (NICs) by the manufacturer for identification.

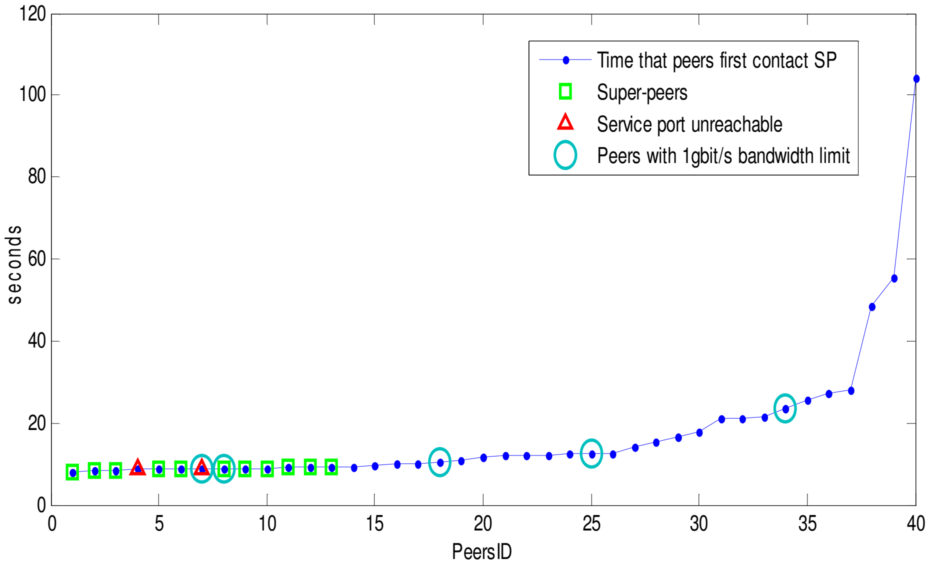


Figure 4.8: The registration time of peers with pre-defined bandwidth limit.

Peer ID

From this figure, we can see there is only one super-peer has 1 gbit/s bandwidth limit. In common sense, to select super-peers, peers with higher bandwidth limit more suitable than peers with lower bandwidth limit. But the result of this experiment clearly shows that the SP has not selected the higher bandwidth limit peer as the super-peer deliberate. The only 1gbit/s peer was selected because it contacts the SP earlier than other peers. Otherwise, all the peers with 1gbit/s bandwidth limit should be selected as super-peers. So we concluded, in this network, bandwidth limit has not impact the super-peers selection. And as we found from the result above, the super-peers are peers which contacted the SP first except two nodes which service port unreachable.

From the above analyses, we conclude that, super-peers algorithm in the SopCast system is based on the registration time at the SP. The fist 11 nodes that contact the SP at the beginning of the experiment are selected as super-peers.

4.5 The Degree Distribution of the SopCast Three Layer Network

As we mentioned above, we defined the SopCast system as a three layer system. Thus, we examine the degree topology at the super-peers and peers separately. If there has video blocks transmitted during a time interval, we define a link between the node pair. The *outgoing degree* means the number of children which a peer supports. The size of this network is 50 peers. The number of super-peers is 11, which means the size of super-peer layer is 11, and the size of normal peer layer is 39.

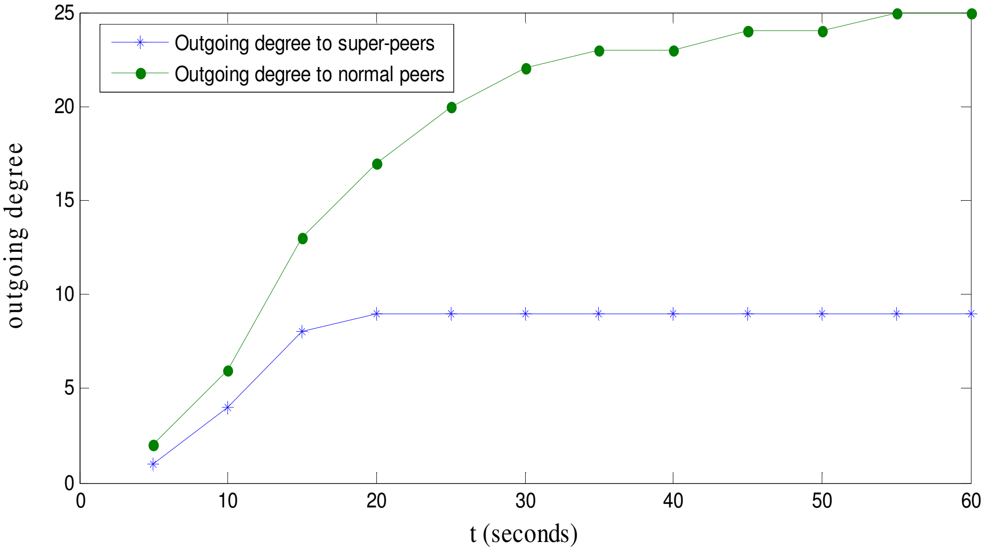


Figure 4.9: History graph of a random peer's outgoing degree.

Previous research [21] shows peers in the SopCast system can discover *neighbors* quite fast. The neighbors they defined are two peers which exchange 42-byte control packets to each other to announce their exits. The neighbor topology already forms to a full mesh network after 300 seconds. In our thesis, we study the topology on super-peers and normal peers separately. But the objective we want to study here is the outgoing degree. That is because not all of the neighbors have video exchange. It happens only between parent-child. We would like to see whether super-peer layer is a mesh network? In Figure 4.9, we randomly select one super-peer, and plot its outgoing degree at a function of time. Because the size of super-peer layer is 11, we can see this peer almost connected all the other super-peers as its children. And time that costs to discovery the other super-peers is quite small. It finished in 20 seconds, and keeps stable after it. But the outgoing degree from this peer to normal peers is keep growing. It connected the majority of normal peers.

In order to understand the whole network, Figure 4.10 is the average outgoing degree of super-peers in this network.

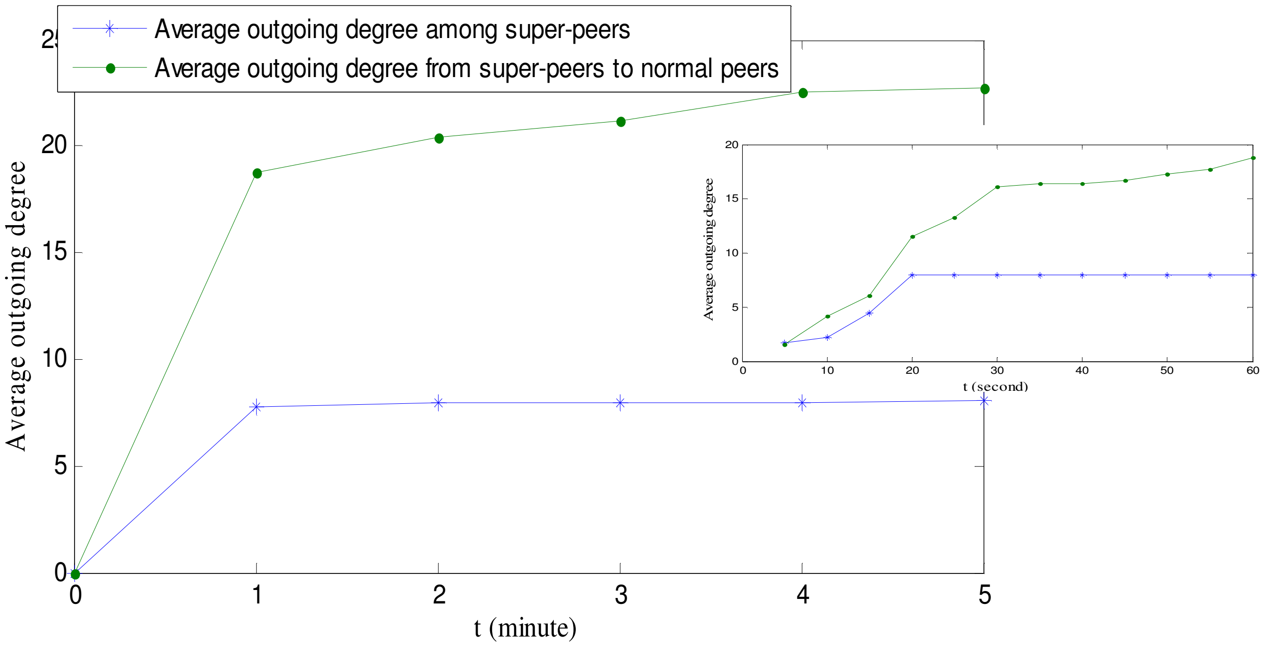


Figure 4.10: History graph of average outgoing degree within the first 5 minutes. The insert figure is the average outgoing degree in the first minute.

We accumulated all the historical link connection in time interval of [0, 5 minute]. We can see the average outgoing degree grows rapidly during the first one minute of the experiment. The first one minute is the most active period for peers to discover neighbors. So we detail the history graph in interval [0, 60 second] in the insert figure of Figure 4.10.

From Figure 4.10, we found the average connection among super-peers is around 8. But the discovery time consumed still very small. We doubt there are several super-peers have low outgoing degree in super-peer layers, which influence the average value. So the straight way is to check the distribution of outgoing degree in the super-peer layers. Figure 4.11 is the distribution of super-peers' outgoing degree within the first minutes.

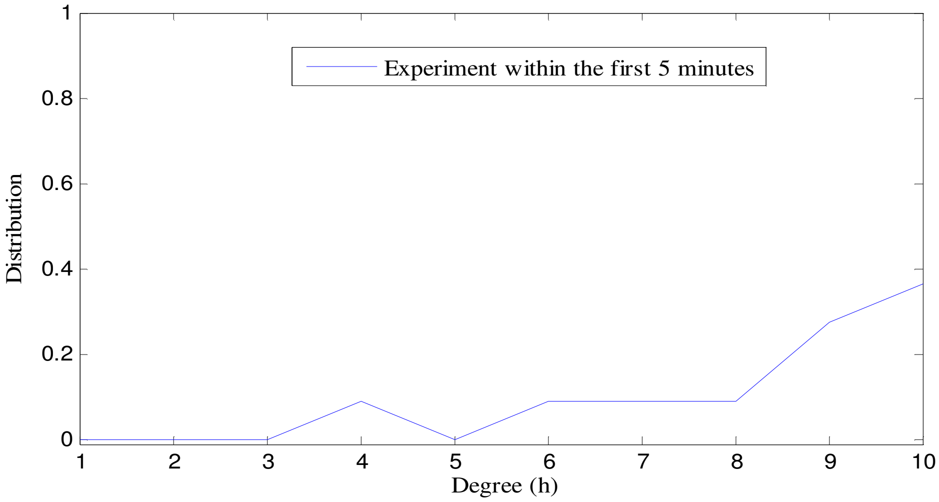


Figure 4.11: Distribution of super-peers outgoing degree within the first 5 minutes.

From Figure 4.11, we can see the super-peer layer is not a full mesh network, but a dense network. That may because the video content of super-peers is mainly download from SP. They need not too many other parents to download with. Which means main responsibility of super-peers is to provide video content to normal peers but not other super-peers.

From the analysis, we knew that the super-peers contribute the major part to supports the services of normal peers. But are the normal peers contribute anything? What is the role of them in the SopCast system? For this problem, we collect the average outgoing degree of normal peers in Figure 4.12. And we divided the outgoing degree from the normal peers to super-peers in this figure and the outgoing degree among normal peers in this figure.

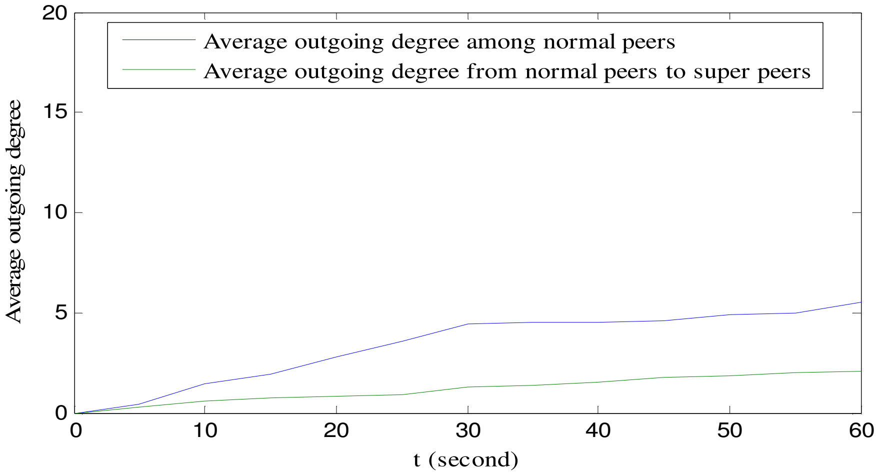


Figure 4.12: History graph of average outgoing degree among normal peers within 1 minute.

From Figure 4.12, we found normal peers also have their own children. Although it grows not as quickly as super-peers, and the super-peers also have larger more outgoing degree than them. But the normal peers have rarely upload data to super-peers. Most of their children are normal peers.

The above analysis helps us to understand the traffic distribution of the SopCast system. We have found that:

- There are a number of peers which download the video content from the SP directly. We define these peers as the super-peers. The super-peers support the remaining peers in the network with video blocks. The super-peers also exchange video content between each other, which forms a dense graph on the super-peers layer.
- The super-peers undertake the main force to upload the video content to support other normal peers.
- The super-peer selection algorithm relies on peer's registration time at the SP.
- The tit-for-tat mechanism of P2P network limits the download throughput of peers who upload few or zero to the network. But in the SopCast, we found normal peers has few contribute to the network, but also can enjoy the high quality services.

Based on the conclusions above, we prefer to study the SopCast network in a dynamic way. We doubt whether the failure of super-peers will influence the stability of the SopCast network? Whether the super-peers changes frequently? What happened if there is new peer join in? In the following chapter, we will study these issues by a series of measurements.

5

The Impact of Peer Dynamics in the SopCast System

5.1 Scenario Description

To study the dynamics of the SopCast network, we have designed the second experimental scenario, which is explained in the following content. A graphic illustration is presented in Figure 5.1.

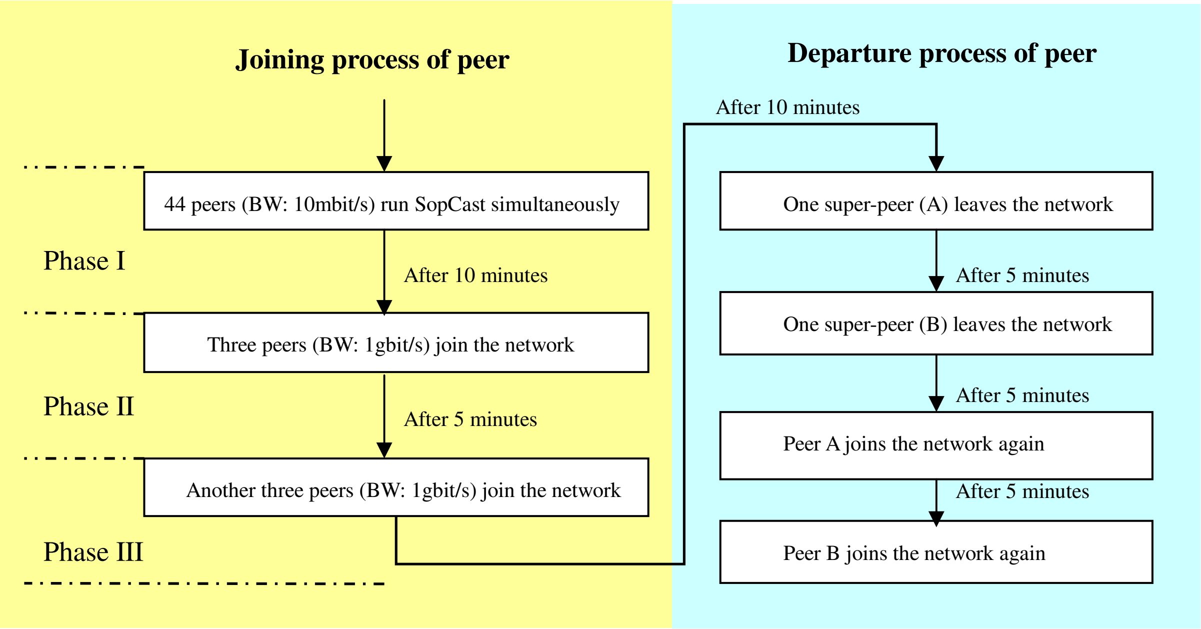

Joining process of peer
Departure process of peer
After 10 minutes
44 peers (BW: 10mbit/s) run SopCast simultaneously
Phase I
After 10 minutes
Three peers (BW: 1gbit/s) join the network
Phase II
After 5 minutes
Another three peers (BW: 1gbit/s) join the network
Phase III
One super-peer (A) leaves the network
After 5 minutes
One super-peer (B) leaves the network
After 5 minutes
Peer A joins the network again
After 5 minutes
Peer B joins the network again

Figure 5.1: An illustration of the second scenario.

In this scenario, we have taken into account the bandwidth of the SopCast peers (PlanetLab nodes). We evaluate the impact of peer dynamics with the peer joining process and the departure process, respectively.

We start the experiment with 44 PlanetLab peers, joining the SopCast network simultaneously. After 10 minutes, we added three peers with 1gbit/s bandwidth to the network. After 5 minutes of that, we added another three peers with 1gbit/s bandwidth to the network. The joining process of peers is thus divided by three phases, as shown in Figure 5.1. Phase I is the period from the beginning to the time that just before we add the first three peers. Phase II started from the time that we add the first three peers, and ended before we add the other three peers. Phase III was from the end of phase II, and finished after 5 minutes.

As we know, the process of peer departure is designed to study the impact of removing super-peers. So we need to know which peers are the super-peers before the experiment of peer departure process. The straightest way is to analyze the traffic of the SP, and list the peers who are supported by SP in this experiment.

The times between the phases are designed base on the results in Chapter 4.5. We have concluded that the peers' average outgoing degree seldom changes after 5 minutes. It increases rapidly in the first minute. This means the network is almost stable after 5 minutes. So after considered the impact of other possible factors, we make phase II happened 10 minutes after phase I to ensure the network is already stable then.

The objective in this scenario is to observe the dynamic of the P2P SopCast system, especially the behaviors of super-peers. In the first peers joining process, we would like to observe whether and how the peers' dynamics impact the super-peers selection. And in peers' departure process, we would like to show the impact of super-peers' changes especially.

5.2 The Impact of Adding Peers to the SopCast Network

As we introduced above, the joining process refers to the procedure of adding peers to the SopCast network during the experiment. We divided the peer joining process into three phases. Figure 5.3 shows each peer's download throughput from the SP in each phase.

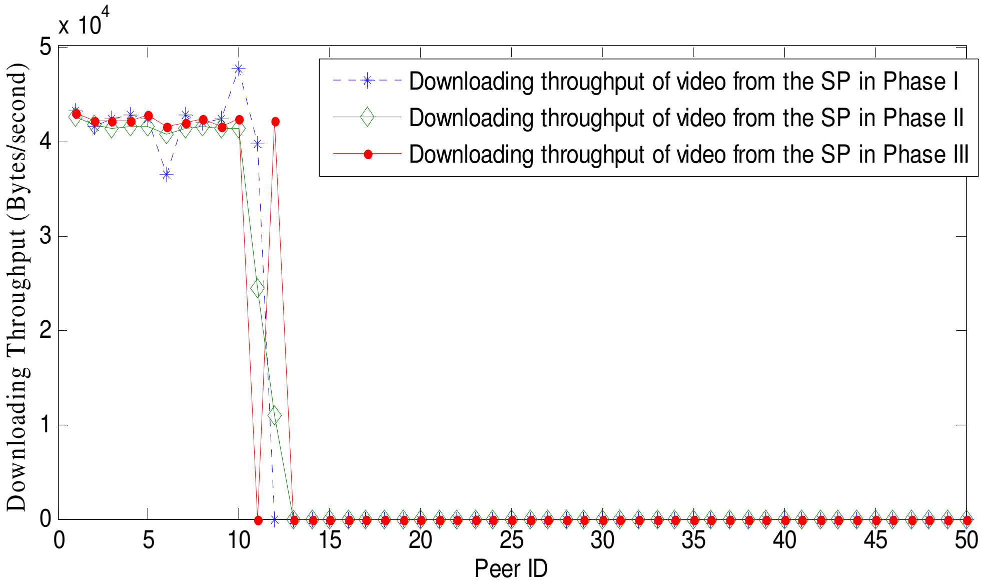


Figure 5.2: Download throughput from SP to peers in each phase.

In Figure 5.2, we can see each peer's download throughput from the SP in different phases. We analyzed the results in this way to study the dynamic of the SP and super-peers when new peers join in the network. In order to check whether there are any changes of super-peers caused by new joining peers, we prefer to compare the throughput in different phases. In Figure 5.2, the peers are ordered with the download throughput from the SP. We can see, the peers in the front part have downloaded video content from the SP during the experiment, and the remaining peers never connected the SP for video downloading. The peers who download video content from the SP directly are the super-peers. Therefore we found that during the whole process, the super-peers did not change except Peer 11 and Peer 12. But the change is not causes by new peers' joining. From the log file of the SP, we found Peer 11 send a 20-byte length packet to the SP during phase II. After that, the SP did not provide video to it any more. This means the SP lost a super-peer (Peer 11) then. In the following traffic of the SP, we found the SP select Peer 12 as the new super-peer instead. And the SP started to provide the video content to Peer 12. That is why in phase II, Peer 11 and Peer 12 all has download throughput from the SP. And the sum of their download volume is almost the same like other super-peers who were supported by SP in the whole period of phase II. This phenomenon enhanced the conclusion in Chapter 4.1.2 that the bandwidth is not the factor of super-peers selection algorithm in the SopCast system. And the super-peers do not change frequently after the network is stable unless the super-peers failed.

The functionality of packets with length 20-byte is unknown for us until now. But in the following experiment, we found that if a peer sends a packet with length 20-byte to one of its parents, this parent will not provide content to it any more. Then, they need to initiate the link again if they want to exchange content. We will introduce this issue in Chapter 5.4.

In the above analysis, we found a peer failed by accident, and SopCast select a new super-peer instead. In the following section, we deeper investigate the super-peers' departure process by removing super-peers.

5.3 The Impact of Super-peers' Departure

From the above analysis, we know that the SP will select a new super-peer instead once one of them failed. But who will be selected as the new super-peer? Does it follow the similar super-peer selection algorithm which is used at the beginning of the SopCast network? The departure process of peers in this scenario is designed to understand these problems.

In the peer departure process, we removed two super-peers sequentially. These two super-peers were randomly selected from the super-peers' list, we refer them as A and B. After we removed the super-peer A and super-peer B, we found that the SP selected the new super-peers. Because of the registration time-based super-peer selection algorithm at the beginning of the SopCast network. We easily considered the contact time as the possible factor to select the new super-peers here. Figure 5.3 and Figure 5.4 are the peers who contact the SP in the following one minute after A/B left, respectively.

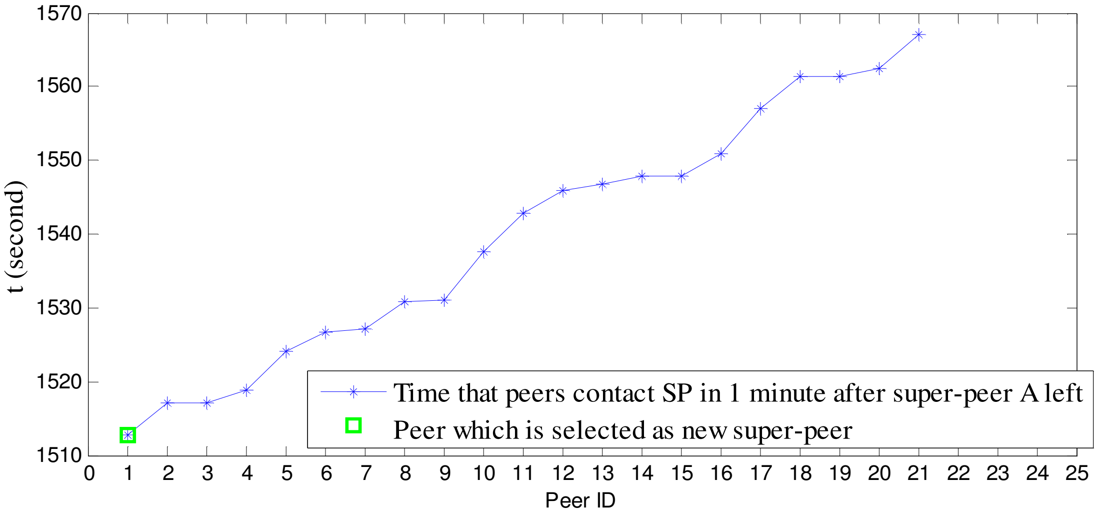


Figure 5.3: Peers contact SP in the following minute after super-peer A left.

The y-axis is the relative time from the beginning of the experiment.

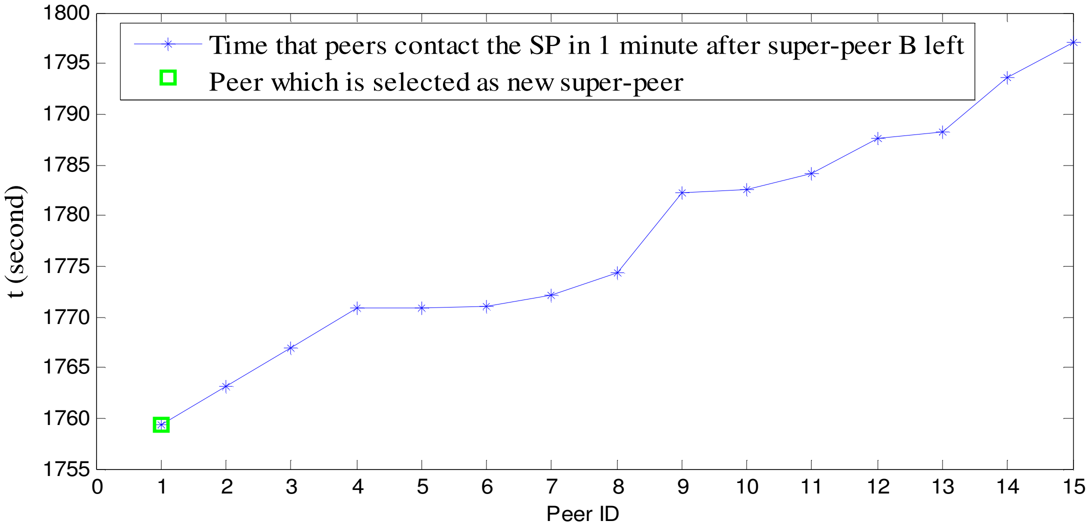


Figure 5.4: Peers contact SP in the following minute after super-peer B left.

The y-axis is the relative time from the beginning of the experiment.

From above two figures, we can see that the SP still selected the peer who contacted it first after the super-peer A left as the new super-peer. And the number of super-peers remains at 11. Therefore we can find that the registration time-based super-peer selection algorithm not only works at the beginning of the experiment, but also through the experiment in case the super-peers' failure. Additionally, we have observed that SopCast reacts promptly to the super-peers failure, and that the registration time-based super-peer selection algorithm copes well with peer dynamics. Registration time seems too simple as the factor of super-peer selection. It is generally regarded that the peer heterogeneity is more suitable as the peer selection factor, such as minimal latency [25] [26] and different bandwidth capacity. But the advantages of registration time-based algorithm are also obviously. It is simple to implement and do not need measurements before peer selection, which means less time consumed in the peer selection process in this kind algorithm. And from the measurement of Fallica [12], SopCast can provide reasonable Quality of Experience (QoE) to peers broadcasting from a PC.

From SopCast's rapid reaction to the super-peers failure, we believe that the registration time-based super-peer selection has the tolerant of network changes. It also implies there should be have communication between normal peers and the SP during the experiment period. So in Chapter 5.4, we prefer to discuss more about the SopCast protocol.

5.4 Discussion on the SopCast protocol

In the SopCast system, if a peer sent a 52-byte packet to another peer, and received the 80-byte reply packet, we refer to the two peers that have established connection [21]. And in our project, we call a peer contact another if it sends a 52-byte packet to another.

5.4.1 Communication between the normal peers and the SP

In the above analyses, we knew that SopCast has the registration time based super-peer selection algorithm. And during the experiment, the SP will select a new super-peer if one of the previous super-peers failed. This new super-peer is the peer who contacts the SP first in the following time. But we doubt are there any other peers communicates with the SP during the experiment except the super-peers. So we collected all the 52-bytes packets from peers to the SP. Figure 5.5 is the number of peers that contacted the SP in each minute.

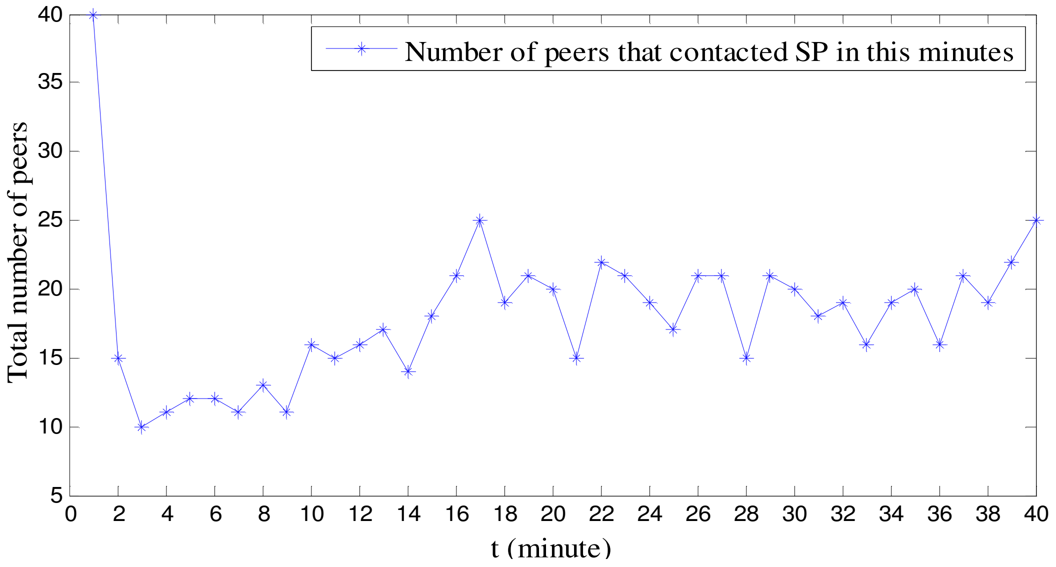


Figure 5.5: The number of peers that has contact the SP as a function of time (in minute).

We can see that at the beginning of the experiment, all active peers have contacted SP (except for the ones that were down unexpectedly). And during the experiment period, around twenty 52-byte packets were sent to the SP every minute. As we know, peers will not send 52-byte packets to peers who already established a link with them. So we believe the peers who contact with the SP in this figure (except the value at the beginning of the experiment) are all normal peers. This conclusion explained why the SopCast system can reactive to the super-peers failure so quickly and without influence of the network. That is because the normal peers keep contact the SP during the experiment period. This makes them available once the SP needs them. But the SP never replies them with 80 byte packets to establish the link unless the SP needs new super-peers.

5.4.2 Number of super-peers in larger-scale SopCast network

During the experiment, we noticed that SopCast has limited the maximum number of children (super-peers) connected to the SP concurrently to 11. It may slightly changes during the experiment, but it must be remains 11 after the network is stable. The experiments in our scenarios involve 50 nodes. The network size may not large enough to present this phenomenon. So we conduct an experiment with 100 nodes to verify this conclusion. Figure 5.6 is each peer's download throughput from the SP in this experiment.

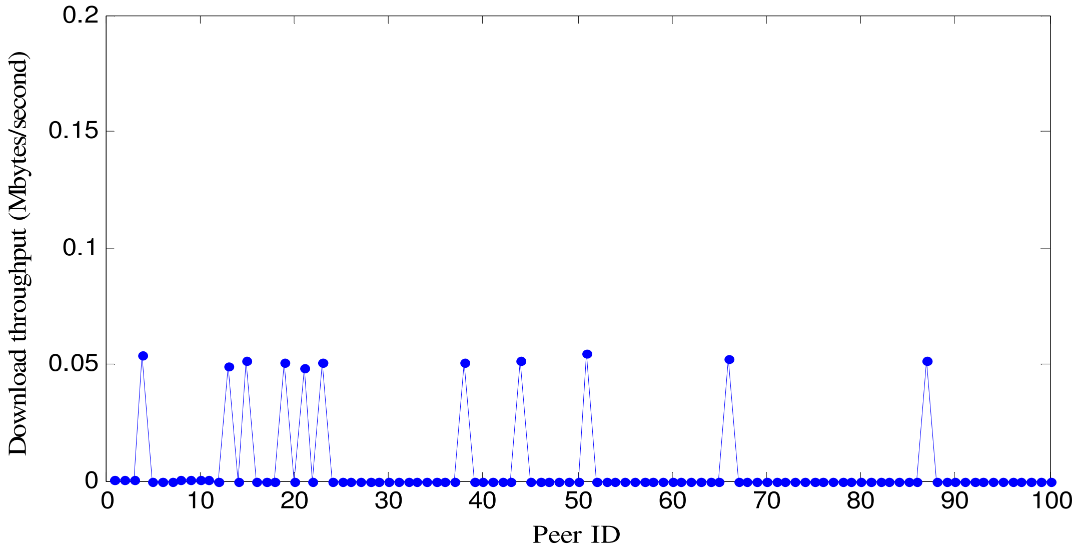


Figure 5.6: Each peer's download throughput from SP (size: 100 nodes).

From Figure 5.6, we can see there are totally 11 peers downloaded content from the SP which means the number of super-peers didn't increased in this 100 nodes network. The number of super-peers is 11, too. So we conclude that even in a larger size of the SopCast network, the number of super-peers in the SopCast system is always limited with 11.

5.4.3 The 20-bytes Packets

As we found above, after a peer sends a packet with length 20-byte to another peers, this parent will not provide content to it any more. They need to initiate the link again by 52-80 byte packets pair when they want to exchange video content again. The 20-byte packets are transferred among peers. We looked inside the log files of each peer, and observed such behavior of each parent-child pair to verify this conclusion.

We collected the packets which were exchanged after sent 20-byte packets, before initiate a link again using 52-80 byte packets pair from each parent-child pair. Table 5.1 is the result of it.

Table 1. Packets transferred after sent 20-bytes packets, before initiate a link using 52-80 bytes packets of each node pair.

Type	Size (bytes)	Percentage (%)
Control packets	52	60%
	80	0
	28	31%
	42	4%
	46	0
	54	5%
Video packets	1320, 377, 497, 617, 1081, 1201	0

The packets with size 377, 497, 617, 1081, 1201 byte are the video fragments and 1320 byte is the maximum size of video packets [21]. So we treat packets with above sizes as the video packets. We can see from this table that the probability of these kinds' packets is zero. This means there is no video packet transfer in this period, which verified the conclusion above.

The analyses in this chapter focus on the overlay's dynamic changes and the SopCast's tolerant of changes. We found:

- Due to the limit of network, we summarized in the small network, the super-peers will not change even there have new peers joined.
- The normal peers keep contact the SP during whole the experiment period. If a super-peer fails, the peer that sends the 52-byte control packet to the SP earlier than other peers will be chosen as the new super-peer.
- SopCast performed good robustness in the experiments. The peer's behavior, even super-peer's failure have not influence the stability of the SopCast network.
- The total number of super-peers in a SopCast system at the same time keeps 11 after it is stable.
- The 20-byte packets exchanged between peers indicate termination of the video link. Peers need to initiate the video link with 52-80 byte packets pair if they want to exchange the video content after that.

6

Conclusion and Future Work

6.1 Conclusion

The main objective of P2P technology is to support a scalable network without a large number of infrastructure and abundant bandwidth resources. The efficiency of the data delivery impacts the network performance significantly. The network will achieve the largest capacity and best performance if every user in the network becomes seed to contribute their resources. But no one application achieved this level yet. Most of the peers in P2PTV network now are free riders. This means that the research on the traffic pattern and operational mechanism of P2P network is important for us to improve the data delivery. But due to the fact that most applications are proprietary systems, there is no source code available. The way we used to understand the mechanism and algorithm for these P2PTV applications are through series of experiments.

Our experiments and analysis mainly focus on the overlay network, which formed by peers. We try to model the traffic pattern and study the super-peer selection algorithm. We designed two scenarios for our objectives: the relative stable network, and the dynamic SopCast overlay. Our results show:

(1) In the SopCast system, there are a number of super-peers who connected with SP directly. They act as the major force to provide video blocks to other peers. So the SopCast network can treat as a three layer network (see Figure 4.2). This three-layer structure only regards the video transmission. The numbers of super-peers are controlled at 11.

(2) SopCast uses the registration time-based super-peer selection algorithm. The SP selects the peers who contact it first as the super-peers. The super-peers rarely change unless they fail unexpected or leave the network. If a super-peer left or failed, the SP will select a new peer as super-peer instead. The new super-peer is the peer who contacts the SP first after the previous super-peer left. If a super-peer leaves the network and joins to it again, it is treated

as a normal peer.

(3) It seems that the fairness mechanism is not implemented in the SopCast network. The download of each peer is uniformed distributed, but the upload is distinctly different. If tit-for-tat protocol is implemented in SopCast, peers who have seldom or no contribution to the network should not download so much.

(4) Peers keeps contact the SP during the experiment period except the super-peers. And if a peer sends a packet with 20-byte to one of its parents, the video link will be terminated. They need to initiate a link again if they want to exchange video content.

The popular way to design a peer selection algorithm is based on their bandwidth capacity, underlying topology or randomly selected. We found that, these parameters are not implemented in the super-peer selection algorithm in. It chose registration time on SP as the factor to select the super-peers. It seems that SopCast does not utilize the peers' resources quite well because it has not choose the best suitable peers (like peers with higher bandwidth, lower latency) as super-peers. But the advantages are also obvious. This algorithm is simple to implement and can cope with the overlay changes promptly. And it does not need more complexity operations to measure the properties of each peer in the network for choosing the best peers. The success of SopCast proved that registration time-based super-peer selection algorithm is an efficient algorithm for mesh based P2PTV network.

6.2 Future Work

Our results helped us to understand the traffic pattern and the super-peer selection algorithm of the SopCast system. These are quite useful for us to understand the mechanisms of mesh based network. Tree based system is another wide used model, such as PeerCast [23]. It must be interesting to measure the tree based P2PTV application and do the analysis. To check how such network works and which one is more efficient for data delivery.

The network size of our experiments is not large enough. In the future work, we will extend the network size, e.g. more than 300 nodes, to verify the conclusions in a larger network.

Super-peers are the most important part of a super-peer based P2P network. So the super-peer selection also will influence the performance of the network. We believe that the current super-peer selection algorithm works well. However, will the performance be improved if we use other factors to select the super-peers, e.g. the bandwidth capacity of each peer? The future research should focus on comparing different super-peer selection algorithms.

Improve the fairness control mechanism will enhance the performance of the SopCast network significantly. Now, the load forces are mainly afforded by SP and super-peers. How to design an improved tit-fir-tat protocol is another interesting aspect for P2PTV researches. Garbacki [6] et al. from Delft university of Technology designed an Amortized Tit-For-Tat Protocol which using bandwidth incentive restriction rather than content.

Reference

[1] PPLive *http://www.pplive.com*

[2] SopCast *http://www.sopcast.com*

[3] Xinyan Zhang, Jiangchuan Liu, Bo Li, and Y. S. P. Yum, "Coolstreaming/donet: a data-driven overlay network for peer-to-peer live media streaming," in *Proc. IEEE INFOCOM*, 2005, pg:2102

[4] TVAnts *http://www.tvants.com.*

[5] BitTorrent *http://www.bittorrent.org/*

[6] Paweł Garbacki, Dick H.J. Epema, and Maarten van Steen, "An Amortized Tit-For-Tat Protocol for Exchanging Bandwidth instead of Content in P2P Networks," *First International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2007)*, pg:119

[7] Michael Feldman, and John Chuang, "Overcoming Free-Riding Behavior in Peer-to-Peer Systems," *ACM Sigecom Exchanges,* Vol. 6.1, July 2005

[8] E. Adar, and B. A. Huberman, "Free riding on gnutella," *Technical report, Xerox PARC,* August 2000

[9] PlanetLab *http://www.planet-lab.org.*

[10] Thomas Silverston, and Marie Curie, "Measuring P2P IPTV Systems," *Laboratoire LIP6/CNRS*, UMR 7606

[11] PPStream *http://www.ppstream.com.*

[12] Benny Fallica, Yue Lu, Fernando Kuipers, Rob Kooij ,and Piet Van Mieghem, "On the Quality of Experience of SopCast," *2008 The Second International Conference on Next Generation Mobile Applications Services and Technologies*, 2008, pg:501

[13] Shahzad Ali, Anket Mathur ,and Hui Zhang, "Measurement of commercial peer- to-peer live video streaming," in *Proc. of ICST Workshop on Recent Advances in Peer-to-Peer Streaming*, 2006

[14] Jehan-Francois Paris, and Purvi Shah, "Peer-to-peer multimedia streaming using bittorrent," *IEEE International Performance, Computing, and Communications Conference*, 2007

[15] Leonardi, and M. Mellia, "Dissecting PPLive, SopCast, TVAnts," *ACM Conext,* 2008

[16] PlMan *http://www.cs.washington.edu/research/networking/cplane/*

[17] Perl *http://www.perl.org*

[18] Windump http:*//www.winpcap.org/windump/*

[19] Tcpdump *http://www.tcpdump.org*

[20] Traceroute *http://www.traceroute.org*

[21] S. Tang, Y. Lu, J. Martın Hernandez, F.A. Kuipers, and P. Van Mieghem, "Topology dynamic in a P2PTV network," *8th International IFIP-TC 6 Networking Conference,* Aachen, Germany, May 11-15, 2009

[22] CoMon *http://comon.cs.princeton.edu/.*

[23] PeerCast *http://www.peercast.org.*

[24] Tracert *http://www.tracert.org*

[25] Chuan Wu, and Baochun Li, “Optimal Peer Selection for Minimum-Delay Peer-to-Peer Streaming with Rateless Codes,” *13th annual ACM international conference on Multimedia*, 2005, pages: 307 - 310

[26] Wei Li, Shanzhi Chen, and Tao Yu, “UTAPS: An Underlying Topology-Aware Peer Selection Algorithm in BitTorrent,” *Advanced Information Networking and Applications*, 2008 22nd International Conference, yr:2008 pg:539

[27] Zattoo *http://zattoo.com/*

[28] Tribler *http://www.tribler.org/*

[29] LiveStation *http://www.livestation.com/*

[30] Plush *http://www.cs.mcgill.ca/~anrl/projects/pShell/index.php*

[31] Pssh *http://www.theether.org/pssh/*

[32] Andy Bavier, Scott Karlin, Steve Muir, Larry Peterson, Tammo Spalink, and Mike Wawrzoniak, “Operating System Support for Planetary-Scale Network Services,” in *Proc. First Symposium on Networked Systems Design and Implementation*, March 29–31, 2004, San Francisco, CA, USA

[33] Wang, F. Mao, Z. M. Wang, J. Gao, and L. Bush, “A Measurement Study on the Impact of Routing Events on End-to-End Internet Path Performance,” in *COMPUTER COMMUNICATION REVIEW*, 2006, VOL 36; NUMB 4, pages 375-386

[34] A. Sentinelli, G. Mar_a, M. Gerla, L. Kleinrock, and S. Tewari, “Will IPTV Ride the Peer-to-Peer Stream”, *IEEE Communications Magazine*, Vol. 45, Issue 6, pp. 86-92, June 2007.

Abbreviation

P2P	Peer To Peer
P2PTV	Peer To Peer Television
VoD	Video on Demand
QoS	Quality of Service
QoE	Quality of Experience
TCP/IP	Transmission Control Protocol/Internet Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
UTAPS	Underlying Topology-aware Peer Selection Algorithm
RTT	Round Trip Time
WMV	Windows Media Video
PlMan	PlanetLab Experiment Manager
GUI	Graphical User Interface
SP	Source Provider
QS	Quality of Source
QC	Quality of Client Network
ICMP	Internet Control Message Protocol
NTP	Network Time Protocol
EST	Eastern Standard Time
SSH	Secure Shell