

國立交通大學

資訊科學與工程研究所

碩士論文

一個在非對稱網路環境下改善
同儕式網路電視播放品質之方法

An Efficient Approach for QOE Improvement of P2P IPTV in
Asymmetric Network Environment

研究生：簡嘉瑋

指導教授：陳耀宗 教授

中華民國一百年八月

一個在非對稱網路環境下改善同儕式網路電視播放品質之方法

An Efficient Approach for QOE Improvement of P2P IPTV in Asymmetric Network
Environment

研究生：簡嘉瑋

Student : Chia-Wei Chien

指導教授：陳耀宗

Advisor : Yaw-Chung Chen

國立交通大學

資訊科學與工程研究所

碩士論文

A Thesis

Submitted to Institute of Network Engineering

Department of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

August 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年八月

學生：簡嘉瑋

指導教授：陳耀宗 博士

國立交通大學資訊科學與工程研究所

摘 要

隨著網路頻寬的普及，越來越多的網路應用興起，而這些技術有部份採用了同儕式網路(Peer-to-Peer Network)架構。不同於以往的用戶伺服器(Client-Server Model)架構，每位使用者(user)將可同時扮演兩種角色，藉以提高整體網路的效能。隨著早期應用如檔案傳輸(File-Sharing)、網路電話(Voice over Internet Protocol)等成功的普及，近年來網路電視(Internet Protocol Television)開始採取同儕式網路(Peer-to-Peer Network)架構，其中最常見的兩種方法分別為：Tree-Push 和 Mesh-Pull，尤其以後者更是已經產品化，最近這幾年，結合 Tree-Push 和 Mesh-Pull 優點的 Push-Pull 機制興起。

當一個或多個同伴離開網路時，傳統的隨機方法將會根據事先定義的標準作業程序和關於這些同伴的資訊來任意的挑選一些代理者(Successor)，這樣一來，將會導致恢復的轉換時間(Recovery switching time)相當冗長，即必須等待長久的時間在於當一個或多個同伴離開後，代理者的尋找。

在本篇的論文中，我們提出了一個和傳統隨機方法比較起來，可以大幅的減低恢復的轉換時間的方法。主要的概念是從同區域的本地端選取事先已經定義好的伙伴做為暫時的代理者，直到系統從伺服器端得到一個固定的代理者。根據同伴離開的方式，我們將它分為規律和不規律兩種模式，在方法上有些差異。另外本篇論文考慮了非對稱網路環境(Asymmetric network environment)，並充分使用所有伙伴的上傳頻寬，而非只有利用高上傳頻寬的伙伴。本篇論文採用 OMNet++ 模擬環境來驗證所提出之方法。模擬實驗結果顯示，我們所提出之方法大幅改善了伙伴離開而造成的播放品質下降問題。

An Efficient Approach for QOE Improvement of P2P IPTV in Asymmetric Network Environment

student : Chia-Wei Chien

Advisors : Dr. Yaw-Chung Chen

Institute of Computer Science and Engineering

National Chiao Tung University

ABSTRACT

As network bandwidth increases quickly and Internet service becomes popular, more and more network applications use peer-to-peer model (i.e. P2P network) model to provide services. Comparing to earlier client-server model, every P2P network user can play two roles, the client and the server. Based on this model, the whole network performance could be improved significantly. As the success of P2P file-sharing and Voice over Internet Protocol (VoIP), more and more IPTV (Internet Protocol Television) employ P2P model to improve their performance, in which, there are two main approaches, tree-push and mesh-pull. The latter has already been deployed widely in the market. In recent years, the combination of these two approaches is emerging.

In traditional random approach, tracker servers will arbitrarily choose one or more peers as successor partners via standard operating procedure with the information of the departed peers, this way will significantly cost long recovery switching time (the time duration from peer leaving to the system recovery).

In this thesis, we propose a method that significantly reduces recovery switching time comparing to traditional random approach. The main idea is to choose one or more pre-defined partners as temporarily successor partners from local partnership side until it finds one or more stationary

successor partners from tracker servers. The approach has a little difference between normal and irnormal mode. This approach can be applied in asymmetric network environment and to fully utilize all of the peers' upload bandwidth instead of only those peers with high upload bandwidth. Finally, we use OMNet++ simulation tool to evaluate the performance of our approach, and the results show that our proposed method improve the quality of experience significantly.



Acknowledgement

First of all, I would like to thank my advisor Prof. Yaw-Chung Chen for leading me into research area and provides me lots of suggestions and many advices on ideas of this thesis. And I also want to appreciate plenty to the members of Multimedia Communication Laboratory, and all my friends, especially for my girlfriend and my family.



Contents

摘要	I
Abstract	II
Acknowledgement	IV
Contents	V
Table list	VII
Figure list	VIII
Chapter 1 Introduction.....	1
1.1 Overview.....	1
1.2 Motivation and Purpose.....	3
1.3 Organization of the Thesis.....	3
Chapter 2 Background.....	4
2.1 Peer-to-Peer Network.....	4
2.1.1 Introduction to Peer-to-peer model.....	4
2.1.2 P2P file-sharing application.....	4
2.1.3 P2P VoIP application.....	7
2.2 P2P Streaming.....	8
2.2.1 Centralized server architecture.....	8
2.2.2 IP layer multicast.....	8
2.2.3 P2P architecture.....	8
2.3 P2P IPTV Overlay construction.....	9
2.3.1 Tree-based overlay.....	9
2.3.2 Mesh-based overlay.....	10
2.3.3 Comparison of mesh and tree overlay.....	11
2.4 Scheduling and adaptation algorithm.....	12
2.5 Push-pull architecture.....	12
2.6 Introduction to QOE	13
Chapter 3 Proposed Approaches.....	15
3.1 Design Philosophy.....	15
3.2 Main Scheme.....	15
3.3 Summary.....	26
Chapter 4 Simulation and Results.....	28
4.1 Environment and Scenario setting.....	28
4.2 Numerical Results.....	28
4.3 Summary.....	31
Chapter 5 Conclusion.....	37
Reference	38

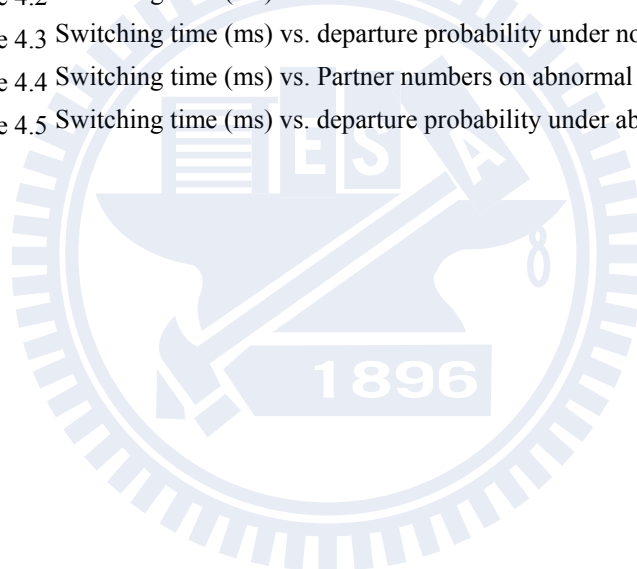
Table list

Table 3.1 System states.....	17
Table 4.1 System parameter setting	31
Table 4.2 Improvement of the switching time (ms) on normal mode.....	32
Table 4.3 Improvement of the switching time (ms) under normal mode with different departure probability.....	33
Table 4.4 Improvement of the switching time (ms) on abnormal mode...	34
Table 4.5 Improvement of the switching time (ms) under abnormal mode with different departure probability.....	35



Figure list

Figure 2.1 Pure P2P model.....	5
Figure 2.2 BitTorrent procedure.....	6
Figure 2.3 Hybrid P2P model.....	7
Figure 2.4 Single tree-based overlay	10
Figure 2.5 Mesh-based overlay.....	11
Figure 3.1 Round robin way on push-pull model.....	16
Figure 3.2 Partner departure for traditional random approach on normal mode.....	18
Figure 3.3 Partner departure for proposed approach on normal mode.....	21
Figure 3.4 Partner departure for traditional random approach on abnormal mode.....	23
Figure 3.5 Partner departure for proposed approach on abnormal mode.....	25
Figure 4.1 Simulation network configuration.....	29
Figure 4.2 Switching time (ms) vs. Partner numbers on normal mode.....	31
Figure 4.3 Switching time (ms) vs. departure probability under normal mode.....	33
Figure 4.4 Switching time (ms) vs. Partner numbers on abnormal mode.....	34
Figure 4.5 Switching time (ms) vs. departure probability under abnormal mode.....	35



Chapter 1 Introduction

Although the P2P (Peer-to-Peer) model has been successfully applied in IPTV (Internet Protocol Television), recently newest push-pull system also face QoE (Quality of Experience) problem. In the rest of this chapter, we will introduce the P2P IPTV system, related work, our motivation and thesis purpose.

1.1 Overview

To realize IPTV system, the most intuition way is client-server model. Client-server model is stable and easy to implementation, but it has lots of disadvantages, for example, hotspot problem, large bandwidth cost, scalability problem and other limitations. On the other hand, P2P model can solve these problems, hence, P2P IPTV become more and more popular.

In P2P model, every peer plays both server and client role, thus greatly reduces the load of servers, and uses peer's upload bandwidth more effectively. The first application of P2P model is file-sharing. Every peer can exchange file like music, pictures or other files with other peers via a common platform. The first famous music sharing is Napster [1]. This P2P platform was very popular after it was published, however, it also faced the copyright problem and quit from the market.

BT (BitTorrent) [2] proposed a new idea that improves the whole P2P system's performance. The key idea is to cut a file into smaller fixed-size chunks in order to download various chunks from different peers simultaneously. This "chunk" term is even used in P2P IPTV system and other application domains.

In addition to file-sharing, P2P technique is also applied in VoIP (Voice over Internet Protocol). Skype [3] is the most popular and well known VoIP platform. This

successful technique can be widely used in conference and other areas.

The newest P2P application is P2P IPTV streaming, such as Raw-Flow [4], Abacast [5], PPLive [6], PPStream [7], UUSee [8], Roxbeam [9], and Mysee [10], with the widespread adoption of broadband residential access and the advances in video compression technologies, IPTV may be the next disruptive Internet application.

The solution of the IPTV can be client-server model, IP layer multicast or P2P model. Client-server model as we mentioned before, has lots of limitation and disadvantage, IP layer multicast has to modify the system of the routers, hence P2P model become the most feasible and most popular solution for the deployment of IPTV service. Two major approaches to construct a P2P system for streaming service:

- Tree-push : Peers form a tree-shaped overlay, and video content is pushed from the originating server to peers.
- Mesh-pull : Peers form a mesh-shaped overlay, and they pull video from each other for content delivery.

The debate on whether mesh-pull or tree-push is more suitable for IPTV is on-going [11]. Tree-push method is still pause in academy area. However, mesh-pull method already become commodity for user to use, such as Cool-Streaming [12], PPLive [13], and PPStream[7].

Those successful commercially deployed mesh-pull systems often suffer from QoE (Quality of Experience) issues, such as start-up delay (from 10 to 30 seconds), playback lag and continuity problem. Hence, in recent years researcher proposed the push-pull method that combine tree-push and mesh-pull method in order to take advantage of these two methods, such as:

- Cool Streaming [14]: the push-pull version of cool streaming.
- GridMedia [15] : tries to overcome the limitation of the DONET pull approach which requests chunk in pull mode at start up and having nodes relaying chunk

in push mode.

- PRIME [16]: the new chunk have been advertised by parents during the last scheduling event and requested by children according to a global pattern of content delivery that minimizes the probability of content bottleneck among peers.

1.2 Motivation and Purpose

When a partner wants to leave the system, the most intuition is to arbitrarily pick up a peer via tracker server as a successor of this partner. This way may cause the same result as above mentioned client-server model. In order to improve this problem and reduce the recovery switching time (from one or more partners leaving to find one or more successor partners), we proposed a method that chooses the previously defined local partner as a temporarily successor. At the same time, we also do the same thing as the traditional random method did, until it finds a successor as a stationary partner, and terminate the chunk transmission from the local temporarily successor partner. This proposed method will significantly reduce the switching time comparing to traditional method, and the improvement increases as number of partner goes large.

In our proposed approach, we will split the problem into normal and abnormal mode. Normal mode means a partner quit application either manually or switching to another channel. On the other hand, abnormal mode means a partner shuts down the system or power fails in unexpected way.

1.3 Organization of the Thesis

The rest of this thesis is organized as follows. First of all, we describe the basic P2P application concept from file-sharing to IPTV streaming in Chapter 2. In Chapter 3, we describe our proposed approach in detail. In Chapter 4, we give a brief description of the OMNet++4.0 tool and evaluate our approach based on the simulation. Finally, conclusion of this thesis is described in Chapter 5.

Chapter 2 Introduction

In this chapter, first of all, we introduce some P2P applications and IPTV systems, tree-push and mesh-pull method and the related newest P2P IPTV push-pull methods.

2.1 Peer-to-Peer Network

2.1.1 Introduction to peer-to-peer model

Earlier in Internet application, client-server is the most popular model, due to its stability and ease of maintenance, but it still has several disadvantages, such as scalability problem, expensive bandwidth, single point of failure problem and bottleneck problem as time goes by and number of users grows. When the user population goes large, the problem becomes more serious.

In order to solve and improve this problem, P2P computing concept are proposed and applied to Internet applications. In P2P model, each peer play both server and client role, that is, every peer can upload and download file to/from other peers in order to find a more efficient way of utilizing peer resource, such as CPU computing power, memory, storage space and upload bandwidth. However, P2P model still encounters some problems which need to be solved or improved. In the following paragraph, we will introduce the P2P application, such as file-sharing, VoIP and IPTV streaming.

2.1.2 P2P file-sharing application

The P2P systems can be split into pure and hybrid model, pure P2P means no any server inside, and all the file transmissions are performed by peers alone. On the other hand, there are several servers that keep metadata information in hybrid P2P model.

Gnutella is the pure P2P model that use flooding method to issue request signal to all of

its connected peers in order to get the file that it wants. Freenet is another pure P2P model example. The pure P2P model is not a good idea, since it waste the Internet bandwidth resource and not very efficient.

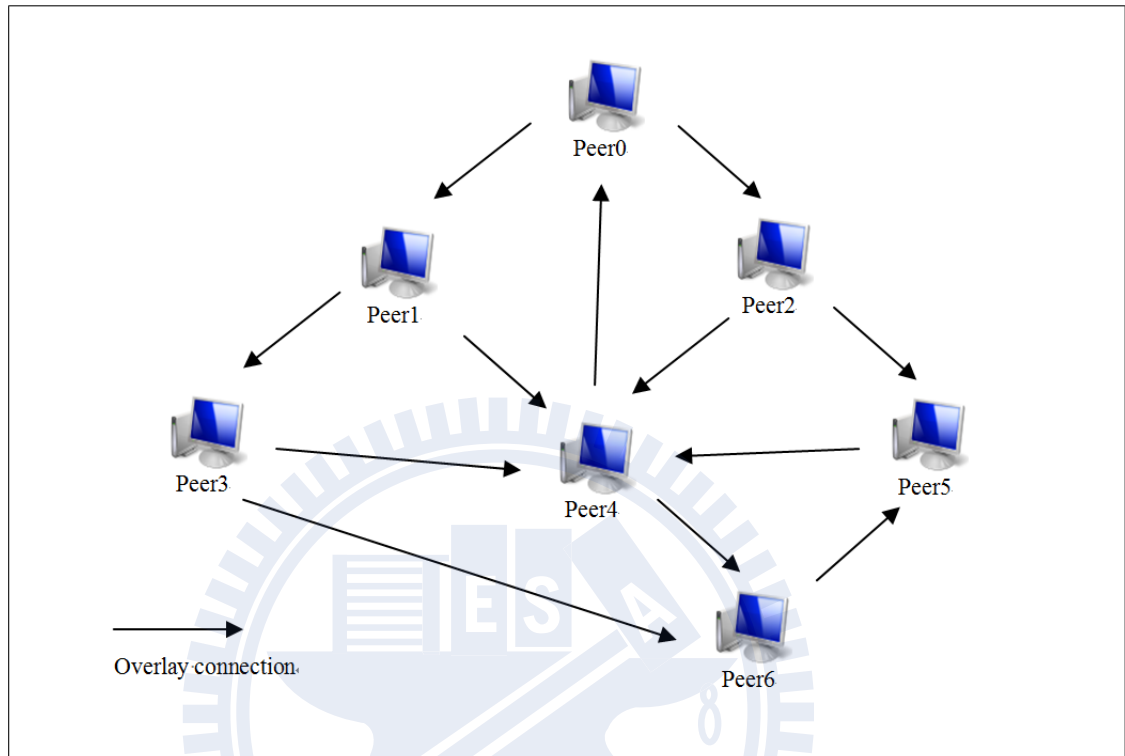


Figure 2.1 Pure P2P model

In Figure 2.1, we can find that there is no any server in pure P2P model, this makes it easy to implement, however, it makes the system hard to maintain.

BitTorrent (BT) was started in 2002, it was a very popular system for file-sharing, and it opens its source code, hence everybody can edit it and create a new different version. In order to deliver the file efficiently, the torrent file is created, this file has metadata information and related tracker servers information, it helps the peer to find out its target file. From torrent files, we can contact a tracker server or several tracker servers to find the target files.

In BT model, every data will be split into smaller fixed size chunk, this made the transmission faster and use bandwidth efficiently. Further, every peer can retrieve

different chunks from different peers simultaneously.

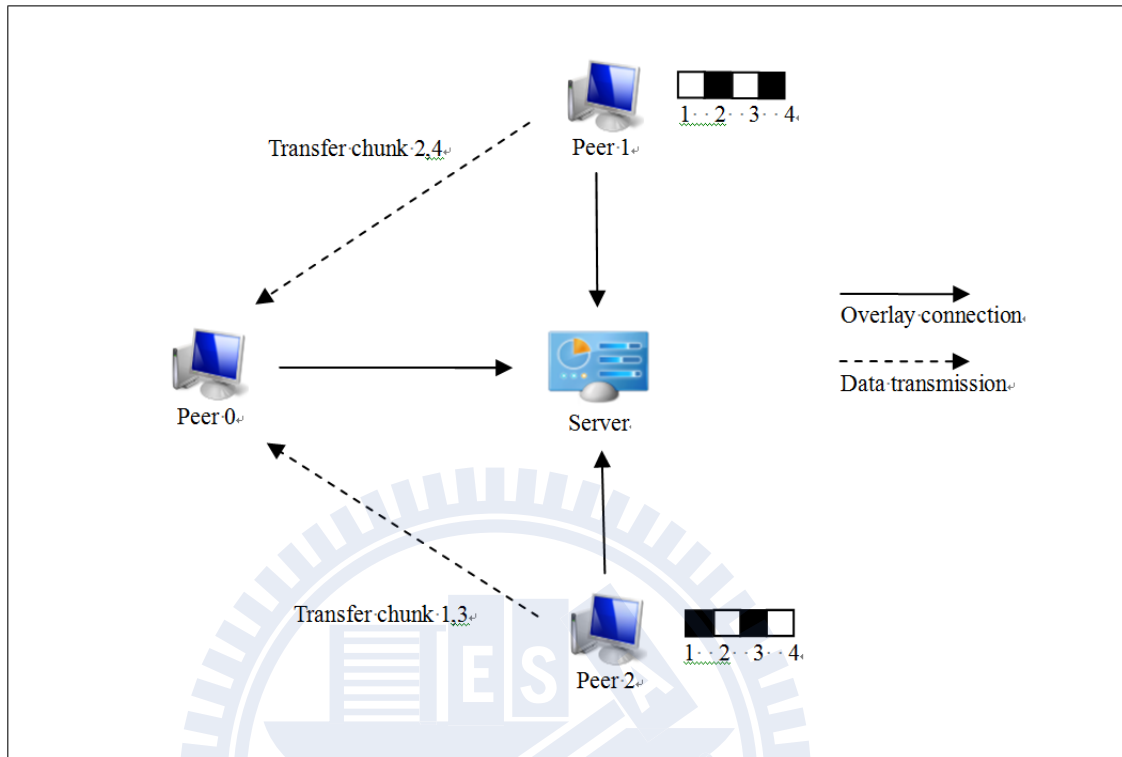


Figure 2.2 BitTorrent operation procedure.

In Figure 2.2, we can understand how a peer retrieves chunk from other peers. First of all, a peer has to find the file torrent which contains some information, such as file name, file size, and related tracker servers IP address. According to this metadata torrent file, peer 0 will issue a request message to related tracker server, which then will send a acknowledgement back to the peer. The acknowledgement contains some information, such as which peers have this file, then peer 0 will issue a request message to peer 1 and peer 2, and peer 1 will send chunk 2 and chunk 4 to peer 0, and peer 2 will pass the chunk 1 and chunk 3 to peer 0. Thus, peer 0 has downloaded all the requested chunks from number 1 to 4 simultaneously.

We can understand the hybrid P2P model concept in Figure 2.3, there are several servers to help P2P system work properly, these servers do the tasks such as storing

every peer's information and keeping the transaction on safety state. We can obtain other information such as which peer has which files from these servers.

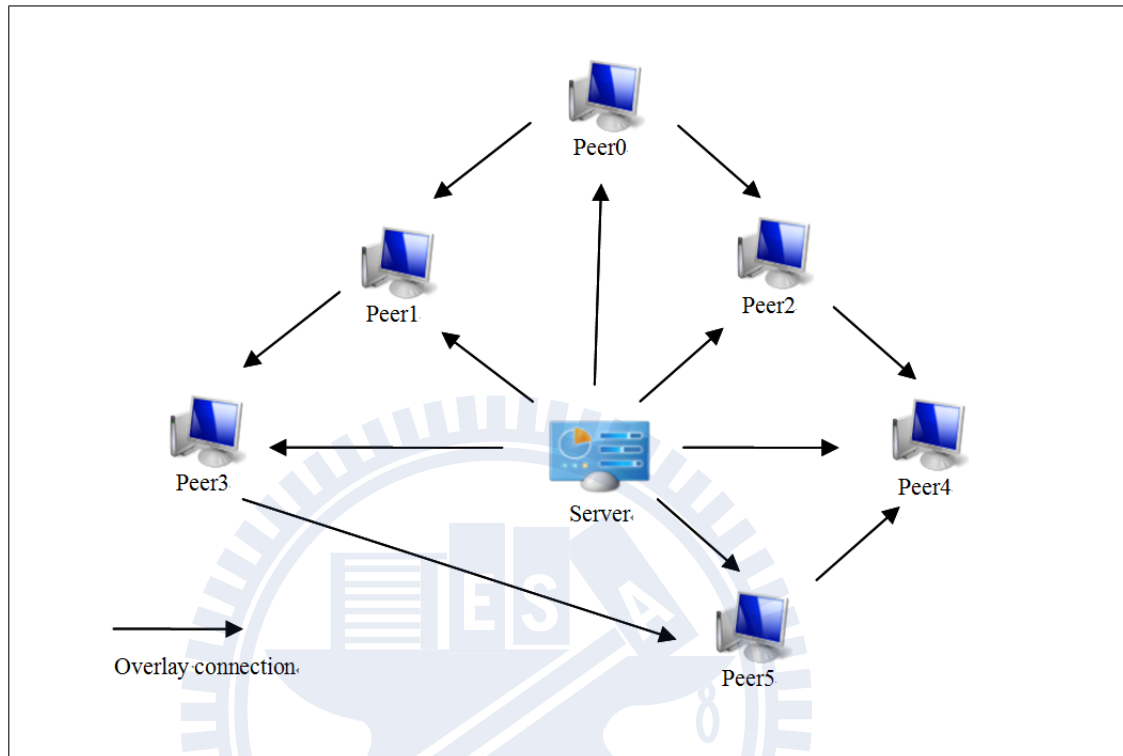


Figure 2.3 Hybrid P2P model.

Another famous hybrid P2P model applied on file-sharing is Napster, Napster is one of the earliest P2P applications. It let everybody share its music through the Internet, however, soon after it started commercial operation, Napster faced the copyright law suit and quit from the market.

2.1.3 P2P VoIP application.

VoIP (Voice over Internet Protocol) is another popular P2P application, and Skype is the one of most wellknown example of the P2P VoIP. Comparing to traditional public VoIP standard such as SIP and H.263, Skype has a well scaling capability. There are several super node in the Skype, every peer depends on and connect to at least one super

node which can be connected to other super-nodes or peers. A peer plays a client role in the beginning, if the peer has powerful capability, then the Skype system will promote the peer to super-node role.

2.2 P2P streaming

P2P streaming is the most popular P2P application recently, user do not need to download the data completely as file-sharing does, hence it solves the free-riding problem, in which the user leaves the system right after it has already downloaded the data. The history of the solution of IPTV system is as follows:

2.2.1 Centralized server architecture

Centralized server architecture is the most intuitive way to realize an IPTV system. However, it encounters the same problem as earlier mentioned client-server model. The video content may even overwhelm the Internet in this architecture, hence it is not popularly deployed.

2.2.2 IP layer multicast

The routers in IP layer multicast method is the key, it is designed to solve potential overwhelm problem and reduce the overhead of the servers in centralized architecture. In IP layer multicast, a server only sends video data with multicast address, then the routers send every copy destined to peers. However, the routers in network core belong to different ISPs, and users (peers) are not authorized to enable this function inside the routers. Hence this method does not work in real world.

2.2.3 P2P architecture

The well accepted IPTV solution is P2P architecture, because it reduces

dramatically sever load, and only increases peer load a little. In P2P IPTV system, every video data is split into smaller chunks, these servers only transfer the video chunks to partial peers, then these peers transfer the chunks to others connected peers. Tree-based and mesh-based are the most of the popular overlay construction in P2P IPTV system.

Overlay construction and scheduling algorithm are the key modules in P2P IPTV system, there are several studies to discuss the issue and propose solutions to achieve better quality of experience (QoE). Membership management, partnership management and peer adaptation are also the necessary functional modules in a P2P IPTV system.

2.3 P2P IPTV Overlay construction

2.3.1 Tree-based overlay

Single tree-based overlay is very easy to implement, every peer deliver the video data to its connected children peers immediately right after it received the data from its parent peers, hence dramatically reduce the delay. However, there is structure reconstruction problem after one or several peers leave. In Figure 2.4, we can clearly realize that leaf peers do not contribute the upload bandwidth.

Multiple tree-based overlays is designed to solve the above mentioned problem, the server divides the stream into several sub-streams for several sub-trees, and these sub-tree peers can deliver the sub-stream to each other. However, multiple tree-based overlays are more complex than single tree-based one in both implementation and maintenance.

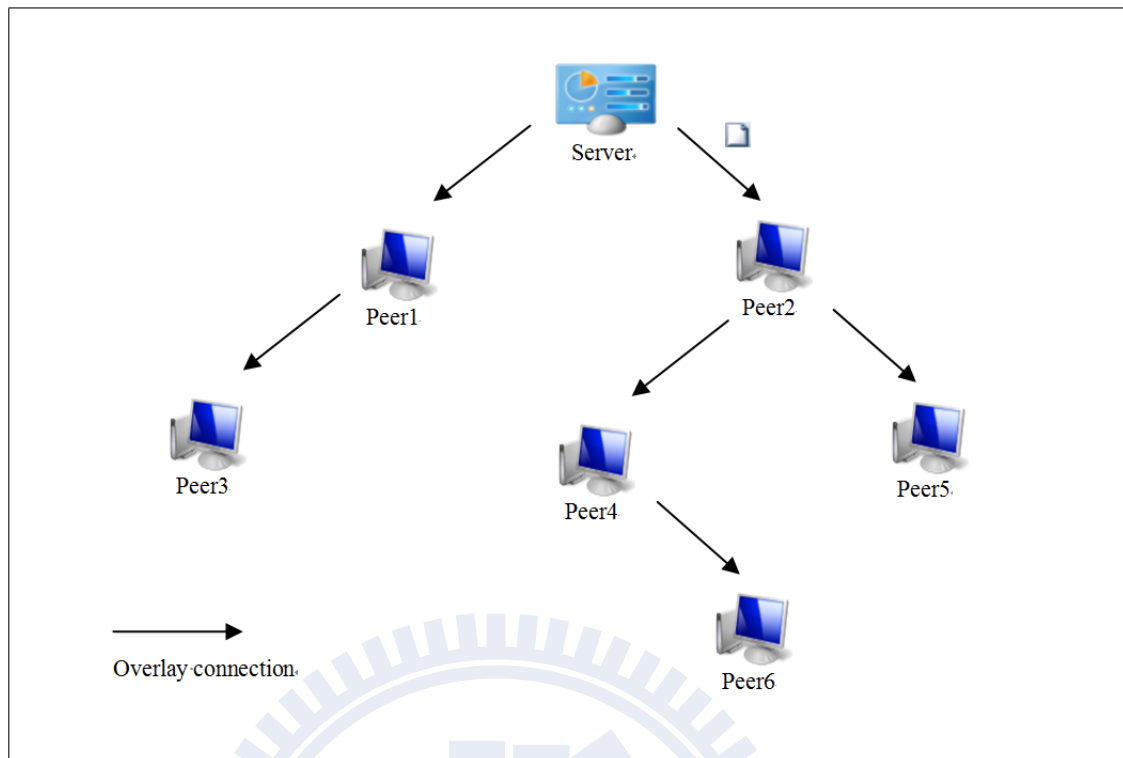


Figure 2.4 Single tree-based overlay.

2.3.2 Mesh-based overlay

As Figure 2.5 shows, in mesh-based overlay, every peer takes a random connection to/from other peers and servers. Hence, unlike tree-based overlay, mesh-based overlay does not need to repair the overlay or reconstruct the structure after one or more peers leave. Pull mode is usually used in mesh-based overlay, that is, every peer issues a request to another partner to receive the video chunk according to the content of the buffer map that is exchanged between partners. Buffer map can tell us which chunks this partner has. Considering a new peer to join the P2P IPTV system, it makes a contact to the tracker server in the beginning to ask for some information, the new peer logs in and selects a channel to watch, then tracker server give the new peer a peer-list that watch the same channel as the new peer does, the new peer chooses some peer from this peer-list as candidate partners. Then the new peer will connect to these candidate partners to make a partnership group. The new peer exchanges buffer map and video

chunks with these partners. Every peer can connect to video server and receive video chunks from it, too.

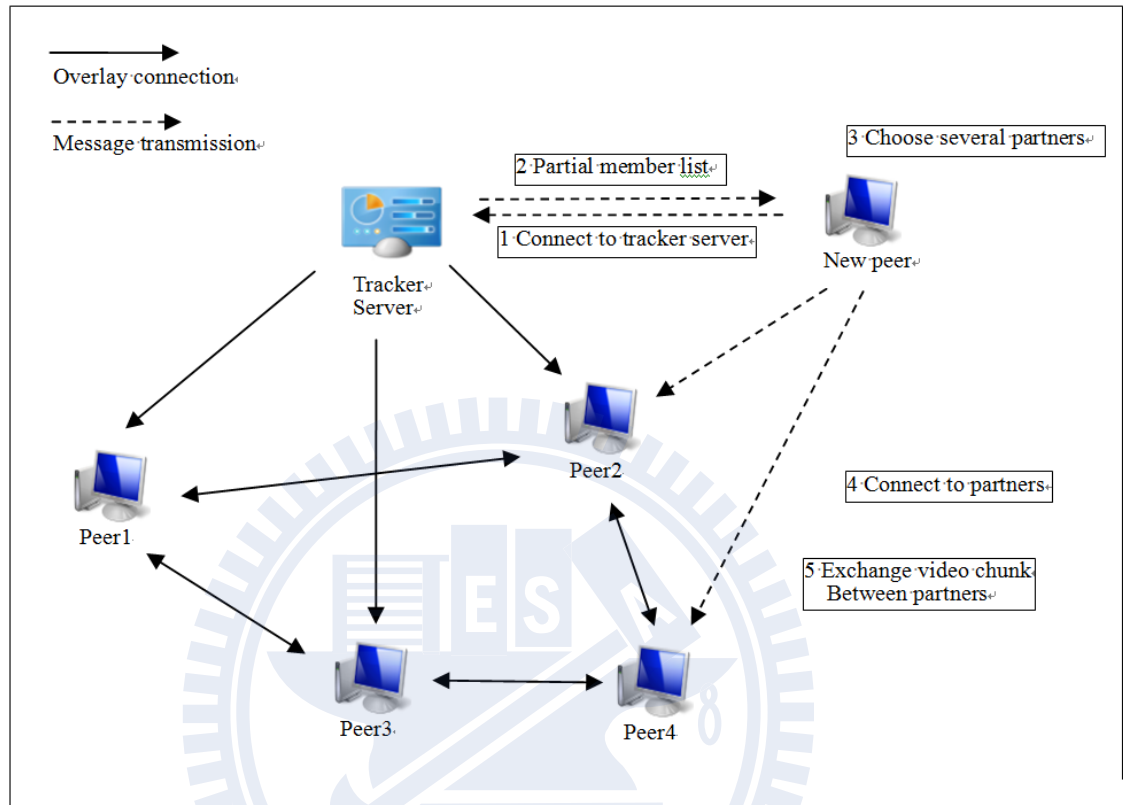


Figure 2.5 Mesh-based overlay.

2.3.3 Comparison of the mesh and tree overlays

There are several studies discussing the difference between mesh-based and tree-based overlay. Unlike mesh-pull systems, tree-push systems have better delay performance when

- The tree structure does not break down due to peer churn, and
- peers at the higher level of the tree have sufficient upload capacity to support the streaming of their children peers.

Tree-push method is still staying in academy research. However, mesh-pull method already has commodity for user to deploy, such as Cool-Streaming, PPLive, and

PPStream. In recent years, researchers take advantage of these two overlays into the new overlay model, push-pull, we will describe the detail in Section 2.5 later.

2.4 Scheduling and adaptation algorithm

Pull and push mode are used in P2P IPTV model, tree-based overlay usually use push mode to deliver streaming data from parent peers to its children peers right after they receive the video stream from their parents. In mesh-based overlay, pull, push and hybrid mode that combines pull and push mode can be used. Usually, mesh-based overlay use pull mode, such as CoolStreaming, PPLive, and PPStream.

Scheduling algorithm tells us which video chunk to send to which partner and which video chunk to request from which partner. Usually random method is used in source peer selection; however, there are several methods for chunk selection, such as the following:

1. Sequential: round-robin way, that is, choose video chunks from head to end.
2. Random: the easiest method to arbitrary pick up a chunk.
3. Rarest-first: to choose the rarest chunks in the local side. Used in CoolStreaming system.

Peer adaptation algorithm keeps the partnership working well. After a long time establishing partnership with other partners and exchange video chunks among partners and source servers, one have to search more suitable partnership for QoE improvement due to some partners leave or network backbone gets broken.

2.5 Push-pull architecture

The current P2P IPTV research is push-pull architecture, it combines the advantage

of the tree-push and mesh-pull architecture. There are several examples, such as:

- Cool Streaming: the push-pull version of cool streaming.
- GridMedia: tries to overcome the limitation of the DONET pull approach which request chunk in pull mode at start up and having nodes relaying chunk in push mode.
- PRIME: the new chunk have been advertised by parents during the last scheduling event and requested by children according to a global pattern of content delivery that minimizes the probability of content bottleneck among peers.

2.6 Introduction to QoE

In P2P IPTV system, we talk about QoE (Quality of Experience) instead of QoS (Quality of Service) normally mentioned in traditional network. QoE focus on customer's perspective to the system that vendor provide, it has several parameters:

1. Start-Up Delay

The time slot between one channel to another channel, the range of time slot usually from 10 seconds to 2 minutes that dramatically longer than traditional television, there are several research paper to improve this problem.

2. Playback Lag

Due to the distance difference among peers, there is playback lag problem in the P2P IPTV system, playback lag means does not synchronization among peer's experience, for example, some peers may watch a home run in live baseball game minutes after the real game.

3. Continuity

Continuity determine whether the video streaming is smooth enough, continuity index (between 0 to 1) can be used to measure, which is the number of video chunks that arrive before or on playback deadlines over the total number of video chunks. If the continuity index arrives 1, that means all of the video chunks have already arrived before the playback deadline.



Chapter 3 Proposed Approaches

Peer's frequent departure is a very common event in P2P IPTV system, when we consider the situation that one or more peers are going to depart from the P2P IPTV system, the current traditional approach that arbitrarily chooses one or more successor partners via pre-defined complexity standard operating procedure with the information of departed partners by tracker servers will cost long recovery switching time (from interrupting the video chunk transmission to the recovery from new successor partners). In the rest of following paragraph chapter, we proposed a method to reduce the recovery switching time and to fully utilize the upload bandwidth of all of the peers instead of high upload-bandwidth peers. Asymmetric network environment (upload bandwidth and download bandwidth of peers is different) is very common in the P2P IPTV architecture, we take it into account and use every peer's upload bandwidth rather than only get chunks from high upload bandwidth peers.

3.1 Design Philosophy

In order to reduce long recovery switching time in traditional random approach, we propose an approach that chooses the local partners as temporary successor partners until it finds steady successor partners via tracker servers. The partner's departure is categorized into normal and abnormal mode (un-expected departure).

3.2 Main Scheme

Before explaining our proposal, we have several assumptions to introduce. First of all, we use push-pull model as our foundation, that is, peers in system form mesh-shape overlay structure and partners each share video chunk by round-robin discipline

simultaneously. As Figure 3.1 illustrates, there are lots of tracker servers who hold the streaming data source and peer-list, such as which peer watches which channel and every partner keeps its channel, available remaining UB (Upload bandwidth), which receiver to send to and which chunk to deliver. These assumption are commonly used both in traditional random and our proposed approach.

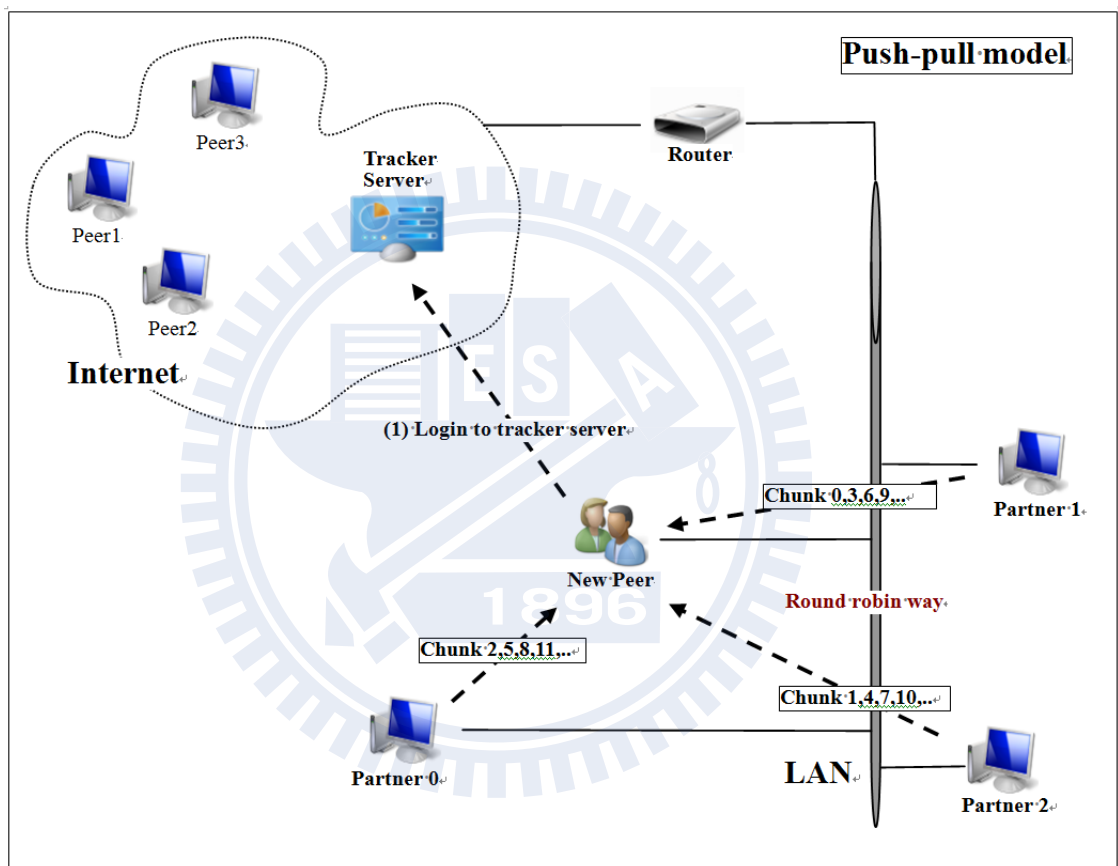


Figure 3.1 Round robin way on push-pull model.

We have extra assumption for our proposed approach, that is, every partner keeps its successor used to replace departed partners, the sequence number of successor can be calculated with three parameters including the sequence number of the departed peer and the number of partners in the following formula:

The sequence number of the successor partner = [(the sequence number of the

departed partner) + 1] modulus (the number of partners).

Every peer has a difference gap between upload bandwidth and download bandwidth, that is, all of peers in asymmetric network environment. Usually, the download bandwidth is bigger than the upload bandwidth of a peer.

Each peer is supposed to reserve certain upload bandwidth (UB) with factor β for a partner departure, the value of factor β will depend on the traffic of whole system, it can be dynamically adjusted by programmer. Every peer remembers its sequence number of parent and corresponding IP address. The sequence number of the video chunks can be calculated as follows:

Table 3.1 The explanation of the terminology

$$\text{Seq[VC]} = \text{Seq[Par]} + \text{Num[Par]} * \text{Round}$$

Notation	Description
Seq[Par]	The sequence number of the partners.
Num[Par]	The number of the partners.
Round	The Bout of the video chunk transmission, R=0,1,2,...
Seq[VC]	The sequence number of the video chunks.

When a new peer joins the push-pull model P2P IPTV system, first of all, the new peer login and register to the tracker server to select a channel to watch. Tracker servers will give it a peer-list which watches the same channel and probably in the proximity of the peer. The upload bandwidth of these peers in the peer-list will close to the download bandwidth of the new peer.

The new peer chooses several peers as candidate partners from this peer-list, and makes connection to these candidate partners. The new peer tells these candidate partners to push the video chunks in a round-robin way.

The departure of partners is very often in most P2P IPTV systems, same in he

push-pull model, however, the traditional random approach cost long recovery switching time, this will significantly effect the QoE (Quality of Experience) of the system.

When a partner departed from the P2P IPTV system, we can categorize this situation into normal mode and abnormal mode. Normal mode means such partner departed from system on common situation that is to quit application manually or switch to another channel. On the other hand, abnormal mode means a partner departed from system on un-expected situation that is to shut down the computer suddenly or the power fails. The following parts have four figures corresponding to four situations for random approach, our proposed approach, normal mode and abnormal mode. Figure 3.2 explains how system recovers when one or more partners departed from the system in traditional random approach under normal mode.

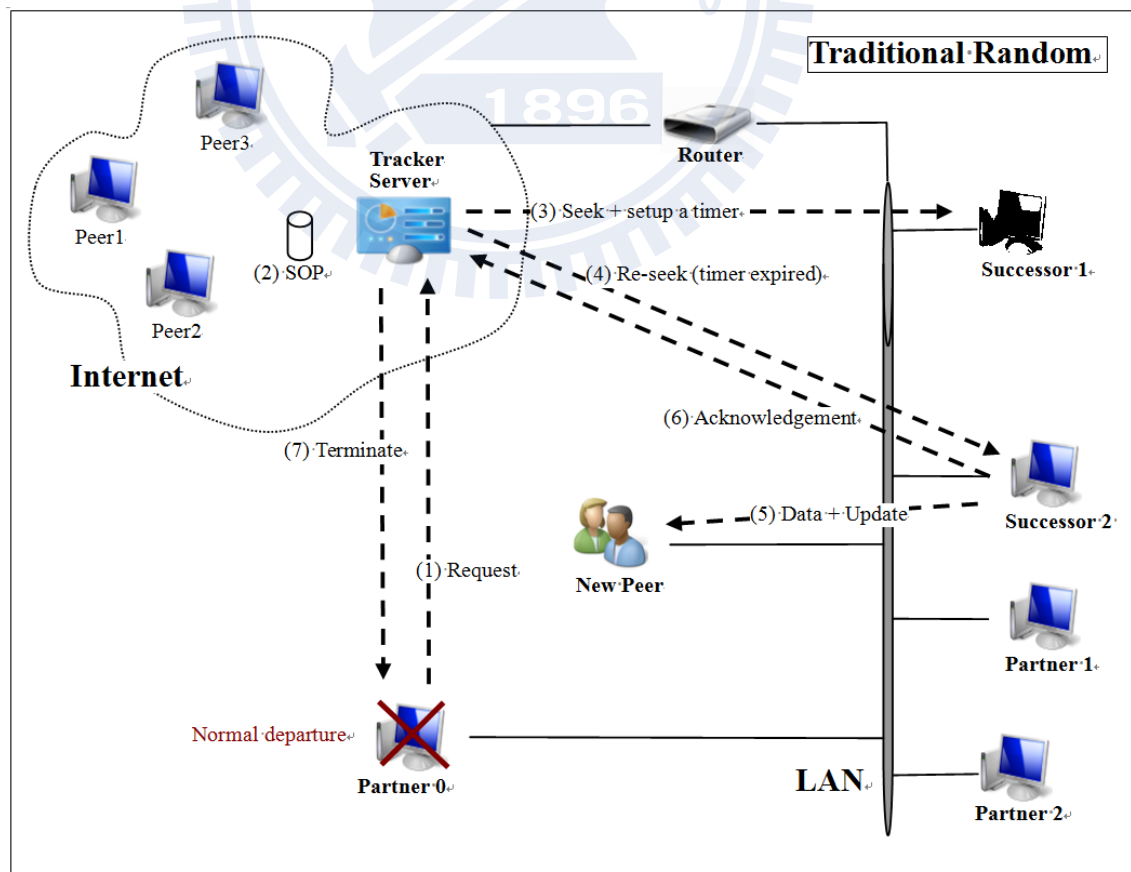


Figure 3.2 Partner departure for traditional random approach under normal mode

As shown in Figure 3.2, when partner 0 would like to depart from the P2P IPTV system, if it is under normal mode, that means it will notify the tracker server to arbitrarily choose a peer to replace this partner as its successor, hence, partner 0 issues a request message to the tracker server with some information such as the sequence number of partner 0, the Internet Protocol address of the partner 0, which channel peer 0 is watching, which chunks are partner 0 supposed sending to, receiver, the number of times, and the number of partners.

Tracker server arbitrarily selects a peer from peer-list to check if it has enough available upload bandwidth, watch the same channel and at least has the next chunk corresponding to partner 0 and adds it to the candidate partner list. Depending on information of peer 0 and the standard operating procedure, the tracker server will establish a sorted queue according to asymmetric network environment (the upload bandwidth of transmitter partners must close to the download bandwidth of the receiver partners) and distance (measured by the IP address of partners) between this successor partner and the new peer (the closer the better), then tracker server will choose the top entry of the queue as successor of partner 0 (named peer s), then move the pointer to the second entry of the queue, and issues a seek message to candidate successors with certain information (i.e. Partner 0 sequence number, next video chunk, receiver (the new peer), partner number, the number of times and so on) and setup a timer, if the timer expired, we think this candidate's successor was gone or has not enough upload bandwidth, then tracker server will re-choose another candidate successor partner from the top of the sorted queue and move the pointer to next entry. If this candidate successor partner (successor 2) exists and has enough upload bandwidth, it will send the next chunk (calculated from the peer d sequence number, partner number and round) to

pass to the new peer as partner 0 normally did, and issues a update message to the new peer in order to refresh the IP address of the partner 0 sequence number which keeps in database of the new peer, then successor 2 issues a seek acknowledgment back to tracker server for reply, when tracker server receive this Ack message, it will issue a termination message to partner 0 to remove it from the system.

Several reasons cause the long recovery switching time in the traditional random approach, the first is to request and seek messages transmission in the long distance between partners in LAN and tracker servers on the Internet (far away from LAN), the second is to establish a sorted queue and doing standard operating procedure, the third is to repeatedly find suitable successors (exist and have enough upload bandwidth) as normal partners.

We proposed a method that dramatically reduces recovery switching time by choosing a local partner as a temporary successor partner until we get the steady successor partner via tracker servers.

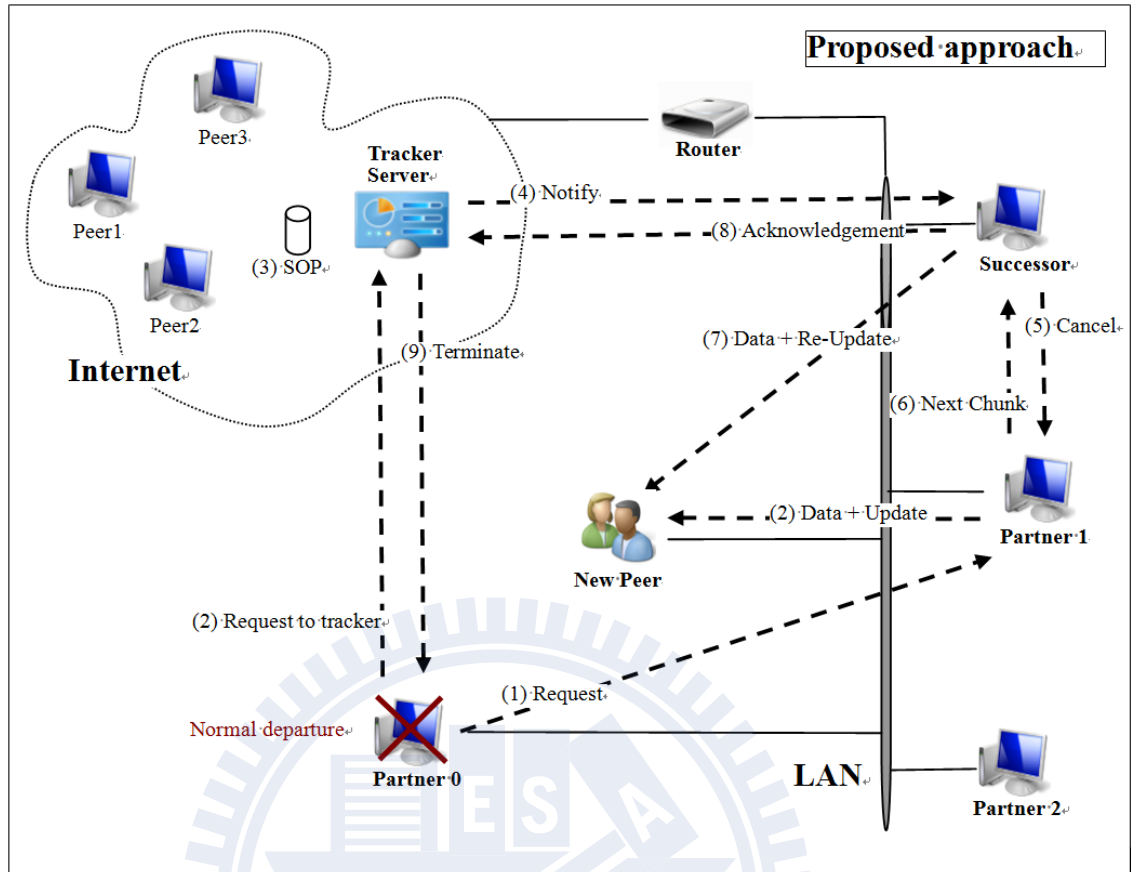


Figure 3.3 Partner departure for proposed approach on normal mode

As Figure3.3 depicted, when partner 0 leave the system on normal mode in our proposed approach, it will issue a request message to partner 1 (the pre-defined successor of the partner 0), then partner 0 issues another request message to tracker servers to find out a steady successor partner. Since we have already reserved each peer's upload bandwidth (the amount of upload bandwidth depend on β factor which is adjusted by programmer and depends on the traffic of the whole system as previously described), the partner 1 definitely has enough upload bandwidth to pass the sequentially video chunks to the new peer by issuing a data message and updates the new peer in order to refresh the IP address of the partner 0 sequence number to be partner 1 which is kept in the database of the new peer.

When the tracker server receives the request message sent from departed partner 0,

it will perform the standard operating procedure and establish a sorted queue with information within the request message (ex. the sequence number of partner 0, the IP address of the partner 0, channel, receiver, next chunk, the number of partners, the number of times and so on), the tracker server issues a notification message (with same information) to the successor (chosen from the top of the sorted queue and move the point to the second entry), the successor definitely has the enough upload bandwidth to pass the sequentially video chunks to the new peer, it issues a cancel message to the partner 1 to get the next chunk sequence number in order to send the sequential video chunks to the new peer. Finally, the successor will issue a re-update message to refresh the IP address of the partner 0 sequence number to be successor which is kept in the database of the new peer.

We can realize that less message and video chunks transmission distance between local partners comparing to long distance between partners in LAN and tracker servers in Internet. Since we have already defined the successor in the beginning, we do not need to find out a suitable successor by complex standard operating procedure or establish a sorted queue, furthermore we reserve certain upload bandwidth adjusted by programming according to the network traffic environment, therefore we can find the successor immediately and definitely.

As we mentioned earlier, partner departure can be spilt into normal mode and abnormal mode. In abnormal mode, partners leave from system in expected way, so we have to do something previously to solve this issue.

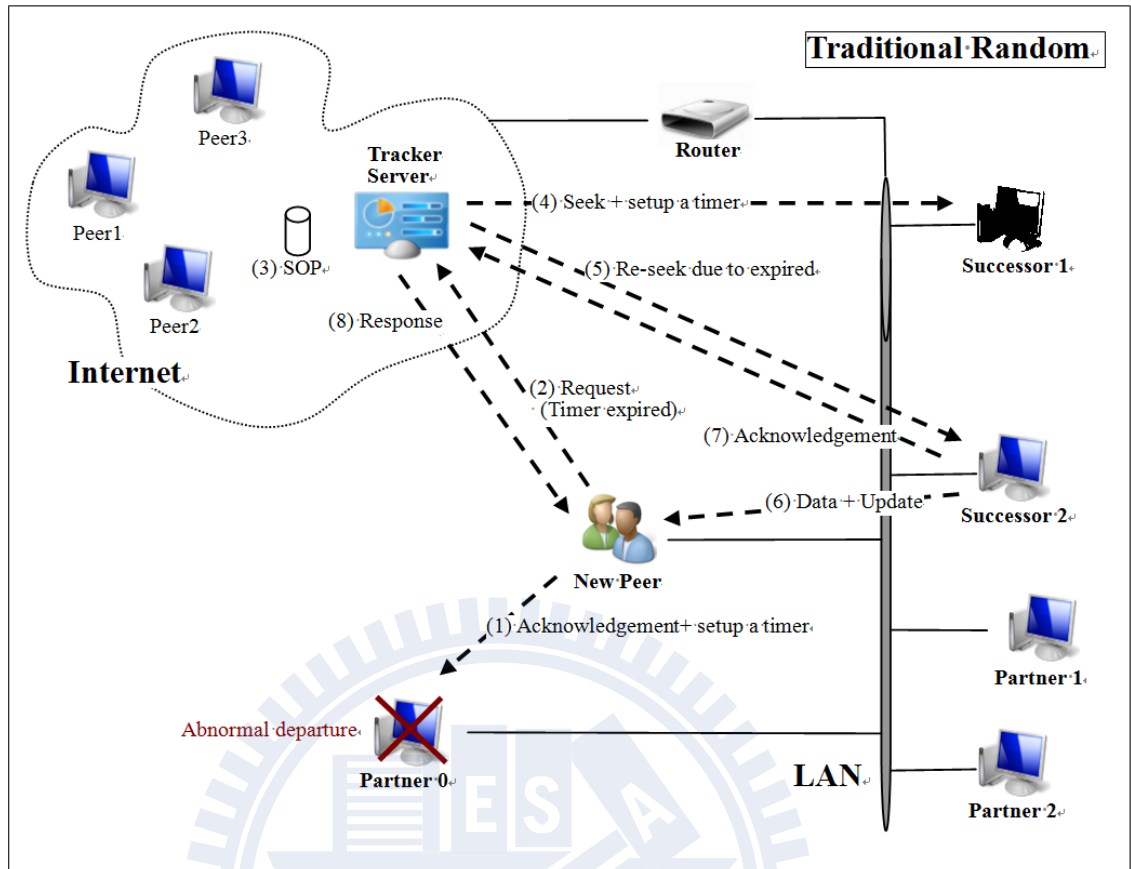


Figure 3.4 Partner departure for traditional random approach under abnormal mode

In Figure 3.4, partner departure under abnormal mode in traditional random approach will end up with that no body knows the departure of partner 0. Hence, every time the new peer received a video chunk from the parent partners and after it sent back an acknowledgment message to the parent, the new peer will start up a timer, if no chunk is received from the parent partners until the timer expires, we assume parent partners was departed from the system, and the new peer will issue a request message with pre-defined certain information (i.e. peer id sequence number, IP address, channel, receiver, next chunk, partner number, the number of times and so on) to the tracker server to search one or more successor partners by carrying out standard operating procedure and establishing a sorted queue with information within request message.

This way will increase delay time comparing to normal mode, because tracker

server can not be notified immediately when the partner 0 departed from the system. If the new peer received a next video chunk from the parent partner before the timer expired, then we terminate the timer and send back an acknowledgment message to the parent partner and re-start up the timer.

The tracker server arbitrarily selects a peer from peer list to check if it have enough available upload bandwidth and watch the same channel and at least have the next chunk corresponding to partner 0 and add it to candidate partner list. When tracker server collects certain amount of information, then based on the information of the partner d and the rule, tracker server builds a sorted queue that is suitable for the new peer from the candidate partner list.

- The closest distance to this partnership (by the IP address).
- The match degree of the download and upload bandwidth.

Then tracker server chooses the top entry of the queue as successor of partner 0, then moves the pointer to the second entry of the queue, and issues a seek message to candidate successor s with certain information (i.e. Partner 0 sequence number, next video chunk, receiver (the new peer), the number of partners, the number of times and so on) and setup a timer, if the timer expires, we think this candidate successor was gone or has not enough upload bandwidth, then tracker server will re-choose another candidate successor partner from the top of the sorted queue and move the pointer to the next entry.

If this candidate successor partner (successor 2) exists and has enough upload bandwidth, it will send the next chunk (calculated from the peer d sequence number, partner number and round) to the new peer as partner 0 normally did and issues a update message to the new peer in order to refresh the IP address of the partner 0 sequence number which is kept in database of the new peer, then successor 2 issues a seek acknowledgment back to tracker server for reply, when tracker server receives this

acknowledgement message, it will issue a response message to the new peer to finish this procedure.

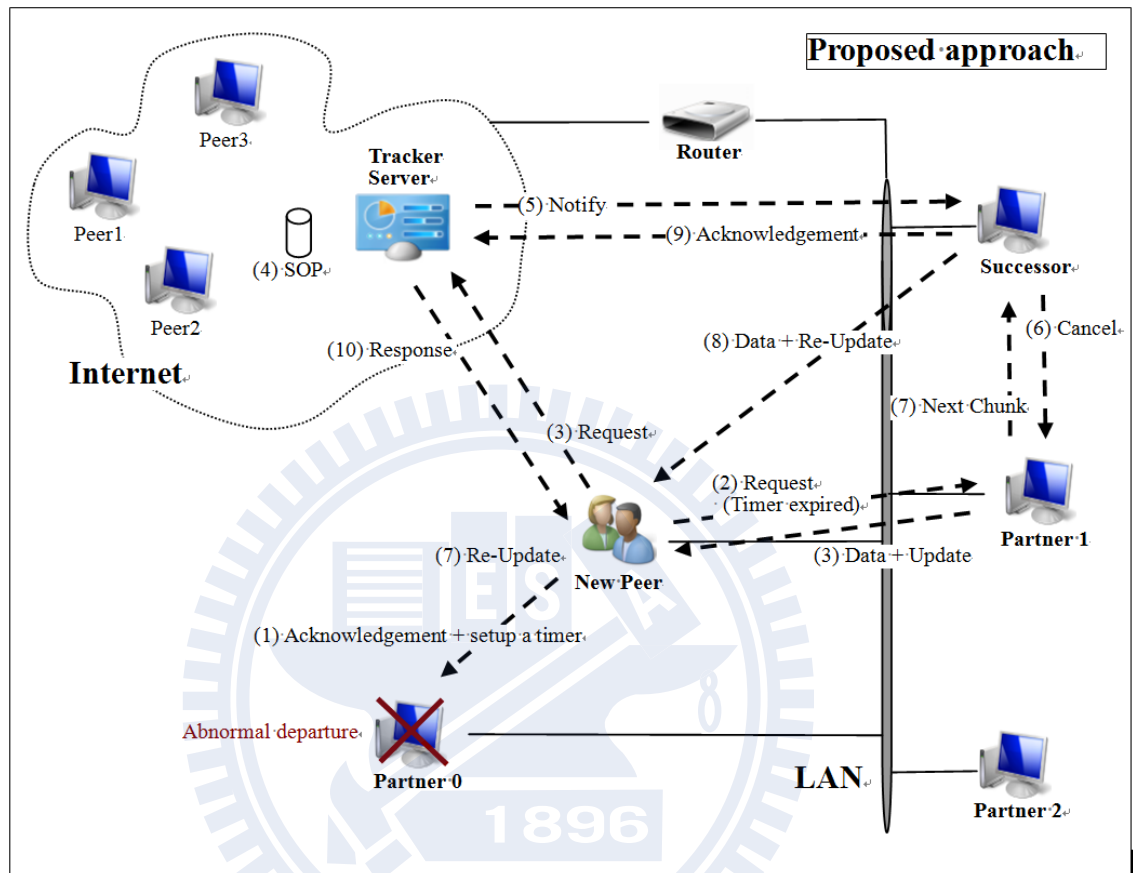


Figure 3.5 Partner departure for proposed approach on abnormal mode.

As Figure 3.5 depicted, under abnormal mode for our proposed approach, every time the new peer send back the acknowledgement message to the parent partners (partner 0), it will starts up the timer, once the timer expired, however, the new peer do not receive any video chunks from this parent partner (partner 0), then the new peer realized that the partner 0 has already departed from the P2P IPTV system. The new peer terminates the timer when it gets a new video chunk, and then resets the timer after sending back an acknowledgment to its parent partners. The new peer calculates out the successor of partner 0 (named partner 1) with several parameters (partner d, sequence number, and partner number) and previously defined formula that already mentioned in

assumption. The new peer issues a request message to partner 1 to replace partner 0. The partner 1 does not need to check its own available upload bandwidth since it already has reserved certain amount of upload bandwidth previously by adjusting β parameter by programmer.

The partner 1 chooses a certain video chunks (calculated based on the partner 0 sequence number, partner number and the number of times) to pass to the new peer as the partner 0 usually did. The partner 1 also updates the new peer by refreshing the IP address of partner 0 sequence number that is recorded in database in new peer.

When the tracker server receives the request message sent from the new peer, it will perform the standard operating procedure and establish a sorted queue with information carried the request message (i.e. the sequence number of the partner 0, IP address, channel, receiver, next chunk, partner number, the number of times and so on), the tracker server issues a notification message (with same information) to the successor (chosen from the top of the sorted queue and move the point to the second entry), the successor definitely has the enough upload bandwidth to pass the video chunks sequentially to the new peer, it issues a terminate message to the partner 1 to get the next chunk sequence number in order to send the sequential video chunks to the new peer. Finally, the successor will issue a re-update message to refresh the IP address of the partner 0 sequence number to be successor which keeps in the database of the new peer.

3.3 Summary

With more and more departure of peers, recovery switching time reduction become more and more important, however, the traditional random approach leads to long recovery switching time, and we proposed a approach to resolve this problem, our method is practical and easy to implement and maintain. The main contribution of this thesis is:

- i. Reduction of the recovery switching time.
- ii. QOE enhancement.
- iii. Reduction of the load of the super-peers.



Chapter 4 Simulation Evaluation and Numerical Results

We evaluate our proposed approach and traditional random approach by using OMNet++ [17] simulator. First of all in this chapter, we roughly introduce the OMNet++ tool and how it works, then we provide parameter situation and show the numerical results by graphical output of our proposed approach and traditional approach.

4.1 Simulation tool

OMNet++ is an extensible, modular, component-based C++ simulation library and framework with an Eclipse-based IDE and a graphical runtime environment. OMNet++ is similar to C++, even more suited to P2P program, but itself does not provide components specifically for computer network simulations, queuing network simulations, system architecture simulation, etc.

4.2 Simulation scenario and parameter setting

In Section 4.2, we provide the simulation scenario and parameters used in our proposed approach and traditional random approach.

4.2.1 Simulation scenario

- Peers form mesh-shape overlay.
- Partners share video chunk by round-robin simultaneously.
- There are multiple tracker servers who hold the streaming data source, peer-list and candidate partner list.
- Every partner keeps its channel, available UB, receiver, chunk_
- Every peer reserves certain upload bandwidth (UB) with factor β .

- Most of the peers leave in normal mode, rest of the peers leave in abnormal mode.
- Proposed approach :
 - Every partner keeps successor, parents sequence number, parents IP address and partner number.
 - The sequence number of the successor partner = [(the sequence number of the departed partner) + 1] modulus (the number of partners).

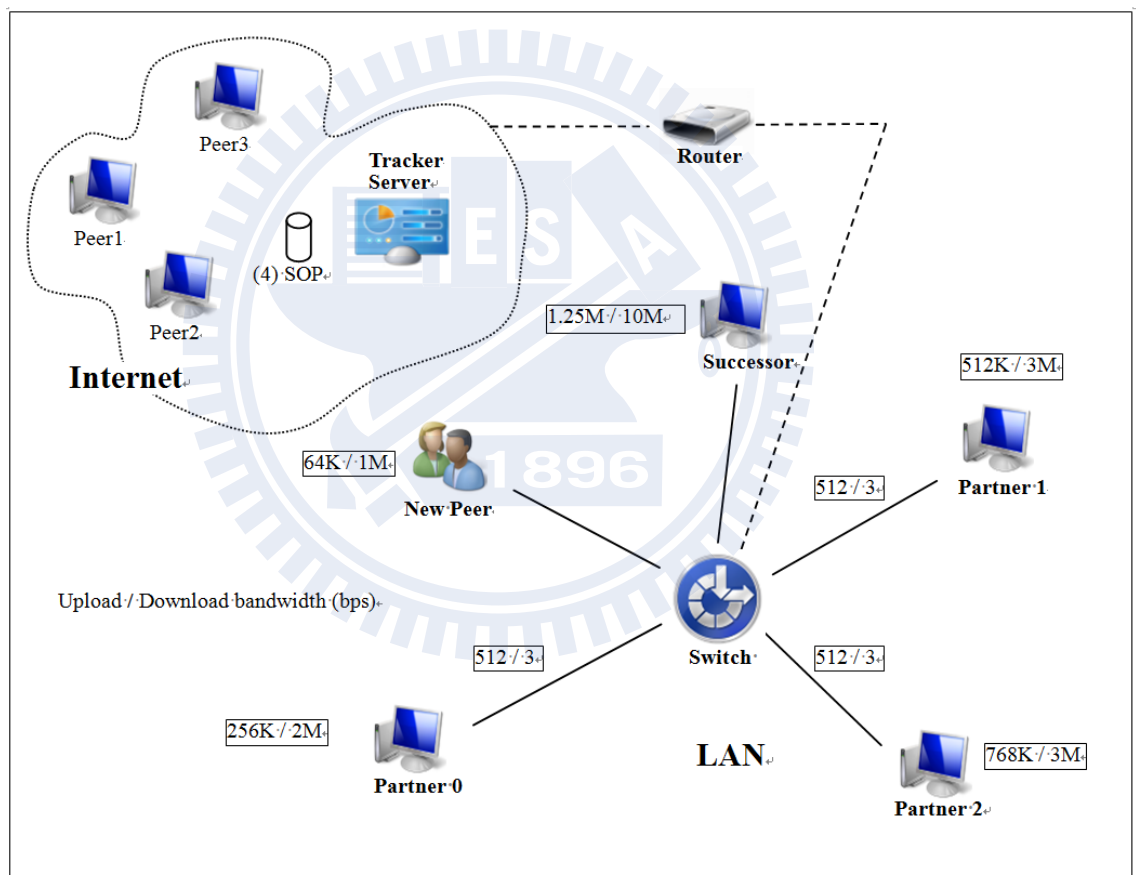


Figure 4.1 Simulation network configuration

Asymmetric network environment in figure 4.1 means most of peers have difference gap between upload bandwidth and download bandwidth, this situation is very common in our network. For our example, partner 0 , 1 and 2 contribute its

bandwidth (video streaming rate / the number of partners, ex: 512Kbps / 3) to the new peer by using round robin way at the same time. This way, the upload bandwidth below video streaming rate still can contribute its bandwidth to transfer video chunks to other partners if it had enough available upload bandwidth.

In our proposed architecture, we use switch to connect every peers, figure 4.1 is the easiest module to implement our method, this architecture can be further expand and scaled to hundreds and thousands peers. The simulation delay and upload / download bandwidth constraint on each peer via OMNet++ tool to setup.

4.1.2 Parameter setting

We can see the parameter setting in Table 4.1, since our proposed approach is push-pull system, hence we use mesh overlay topology, we set our simulation time to 1 hour. Each fixed video chunk is 512Kbits which equals to 1 second of video for streaming rate of 512Kbps. Although higher rate means better QoE, we take lower download bandwidth user into account, and further, 512Kbps is enough for P2P IPTV with decent quality. The range of the number of partners between 3 and 8, the partner's departure probability of partners means the possibility that partners will leave from the system is set from 0.01 to 0.5, and the upload bandwidth varies from peer to peer.

Setting Parameters	Values
Overlay topology	Mesh
Simulation time	1 hour
Streaming rate	512 Kbps
Chunk size	512 Kbits (1 sec)
Number of Partners	From 3 to 8
Departure probability	From 0.01 to 0.5

Table 4.1 System parameter setting

4.3 Numerical results

In Section 4.3, we compare our proposed approach with random approach by running simulation tool OMNet++ 4.0 with above scenario and parameters, and shows the results in graphical way via Microsoft Excel tool.

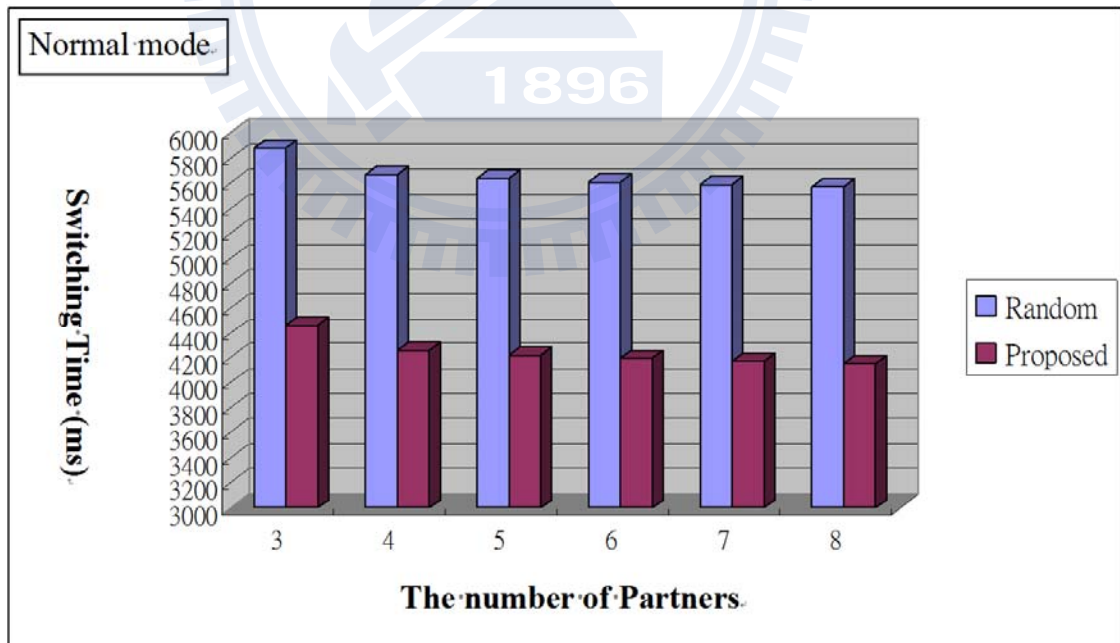


Figure 4.2 Switching time (ms) vs. Partner numbers under normal mode

Normal mode.

Partner	3	4	5	6	7	8
Departure	5141	4931	4896	4869	4847	4830
Recovery (random)	11023	10604	10535	10480	10437	10401
Recovery (proposed)	9607	9188	9118	9064	9021	8985
Partner	3	4	5	6	7	8
Recovery Switching Time (random)	5882	5673	5639	5611	5590	5571
Recovery Switching Time (proposed)	4466	4257	4222	4195	4174	4155
Improvement	24.07%	24.96%	25.13%	25.24%	25.33%	25.42%

Unit: ms.

Table 4.2 Improvement of the switching time (ms) under normal mode

As we observed regarding the degree of reduction of the switching time in Figure 4.2 and Table 4.2, our proposed method dramatically reduces the switching time that started from a partner leaving to a new partner successor transferring next video chunk, the gap between our proposed approach and traditional random approach grows as number of partner grows larger. However, the switching time of both approaches decrease grows as number of partners grows higher.

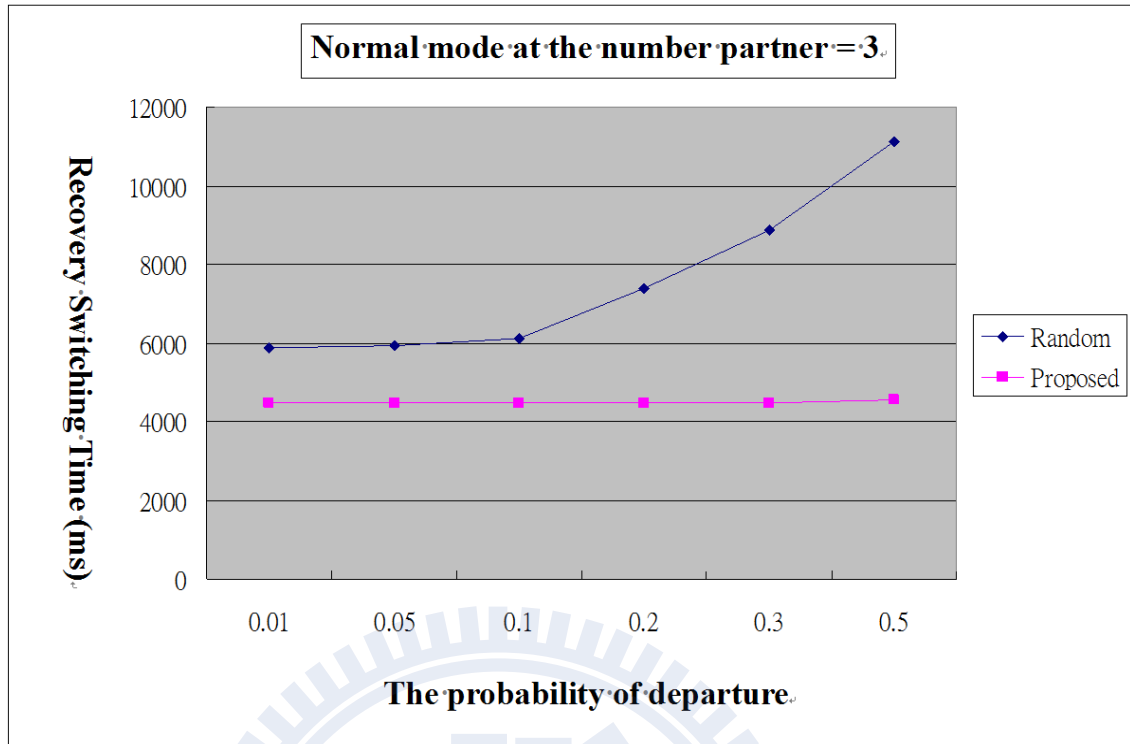


Figure 4.3 Switching time (ms) vs. departure probability under normal mode

As we can see in figure 4.3 and table 4.3, while the probability of the departure of peers grow up, the recovery of switching time of the traditional random method increases dramatically, and thus enlarge the gap from our proposed approach. This is because when more and more peers leave from the system, all of the request will overwhelm the tracker servers, thus the recovery switching time must be put off.

Probability	0.01	0.05	0.1	0.2	0.3	0.5
Departure = 5141 ms at the number of partners = 3						
Recovery (random)	11023	11073	11273	12523	14024	16273
Recovery (proposed)	9607	9607	9607	9607	9607	8985
Probability	0.01	0.05	0.1	0.2	0.3	0.5
Recovery Switching Time (random)	5882	5932	6132	7382	8883	11132
Recovery Switching Time (proposed)	4466	4466	4466	4466	4466	4566
Improvement	24.07%	24.71%	27.17%	39.50%	49.72%	59.88%

Table 4.3 Improvement of the switching time (ms) under normal mode with different departure probability

Table 4.3 shows our experiment with the number of partner 3, and our proposed method

has minor increase when the probability of the departure of peers grows higher.

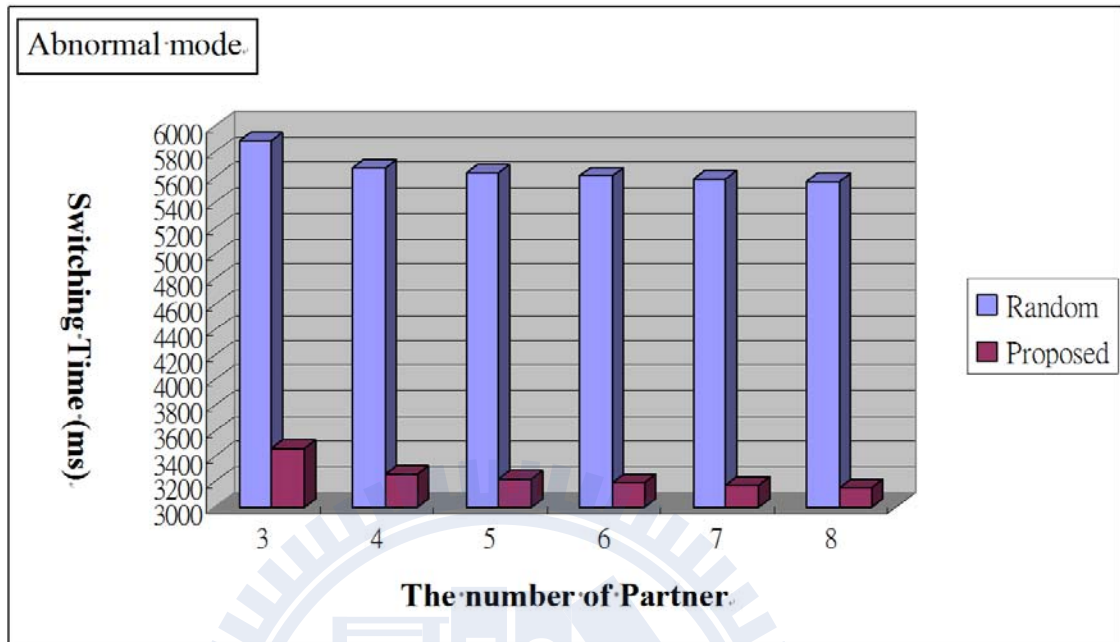


Figure 4.4 Switching time (ms) vs. number of partners under abnormal mode

Abnormal mode						
Partner	3	4	5	6	7	8
Departure	5441	5232	5197	5170	5148	5131
Recovery (random)	11322	10903	10834	10779	10736	10700
Recovery (proposed)	8907	8488	8418	8364	8321	8285
Partner	3	4	5	6	7	8
Recovery Switching Time (random)	5881	5671	5637	5609	5588	5569
Recovery Switching Time (proposed)	3466	3256	3221	3194	3173	3154
Improvement	41.06%	42.59%	42.86%	43.06%	43.22%	43.37%

Unit: ms.

Table 4.4 Improvement of the switching time (ms) under abnormal mode

As we observed regarding the degree of reduction of the switching time in Figure 4.2 and Table 4.3, the degree of improvement in our proposed method dramatically reduces

the switching time much more than that on normal mode. Due to the less message transmission on abnormal mode, the timer expiration is postponed under abnormal mode.

Probability	0.01	0.05	0.1	0.2	0.3	0.5
Departure = 5441 ms at the number of partners = 3						
Recovery (random)	11322	11372	11572	12822	14536	16572
Recovery (proposed)	8907	8907	8907	8907	8907	9007
Probability	0.01	0.05	0.1	0.2	0.3	0.5
Recovery Switching Time (random)	5881	5931	6131	7381	9095	11131
Recovery Switching Time (proposed)	3466	3466	3466	3466	3466	3566
Improvement	41.06%	41.56%	43.47%	53.04%	61.89%	67.96%

Table 4.5 Improvement of the switching time (ms) under abnormal mode with different departure probability

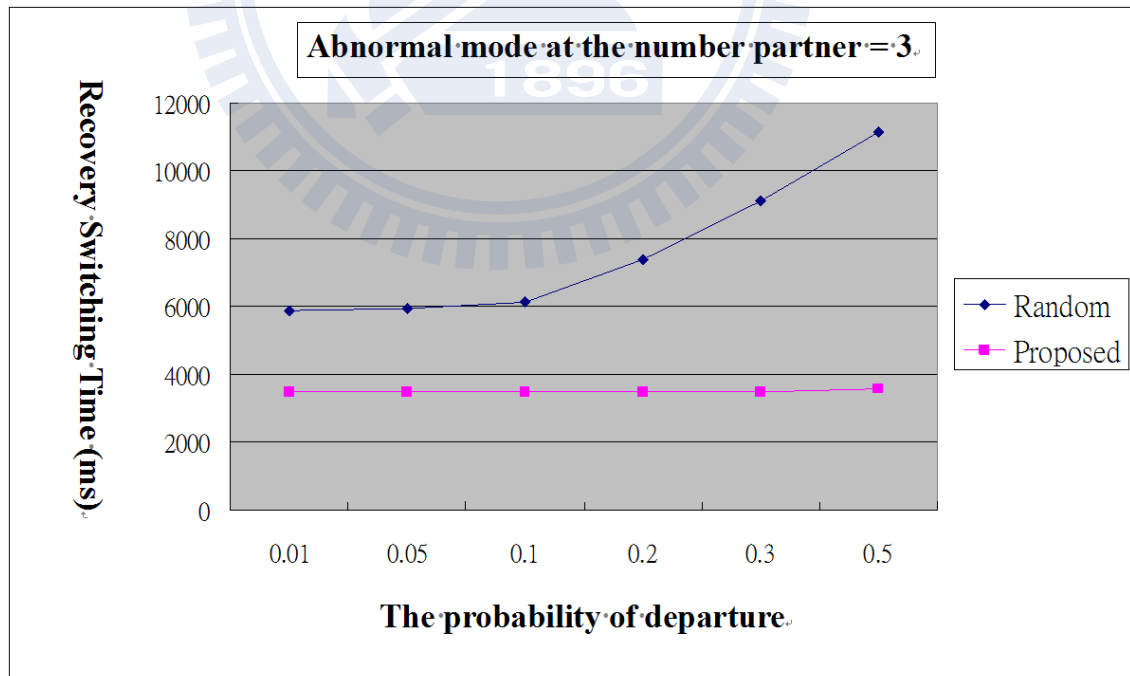


Figure 4.5 Switching time (ms) vs. departure probability under abnormal mode

As we can see in figure 4.5 and table 4.5, while the probability of the departure of peers grow up, the recovery of switching time of the traditional random method increases dramatically, and thus enlarge the gap from our proposed approach on abnormal mode.



Chapter 5 Conclusion

With the widespread adoption of broadband residential access and the advances in video compression technologies, the P2P IPTV has become more and more popular. However, the reduction of recovery switching time problem has become significant due to the frequent churn of peers.

In traditional random approach, recovery switching time is quite long due to the far distance from peers to tracker servers and hence the larger message exchanging delay needed by the complex standard operating procedure even if it is an intuitive way to implement.

In order to reduce the recovery switching time, we choose local partners as temporary successor partners until it finds one or more constant partners via tracker server in our proposed approach.

We use OMNet++ 4.0 simulator tool to evaluate our proposed approach comparing to traditional random approach. The improvement of the recovery switching time of our proposed approach is about 24~60% under normal mode, and 41~67% under abnormal mode. The improvement gap of the recovery switching time between our proposed approach and traditional random approach increases while number of partners grows up.

Our proposed approach is practical and feasible to implement and maintain.

References

- [1] Napster, <http://free.napster.com/>.
- [2] BitTorrent, <http://www.bittorrent.com/>.
- [3] Skype, <http://skype.com/>.
- [4] RawFlow Ltd., URL: <http://www.rawflow.com/>.
- [5] Abacast Online, URL: <http://www.abacast.com/>.
- [6] PPLive.com, URL: <http://www.pplive.com/>.
- [7] PPStream, URL: <http://www.ppstream.com/>.
- [8] UUSEE, URL: <http://www.uusee.com/>.
- [9] Roxbeam media network, URL: <http://www.roxbeam.com/>.
- [10] Mysee, URL: <http://www.mysee.com/>.
- [11] N. Magharei, R. Rejaie, and Y. Guo, “Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches,” *IEEE INFOCOM*, 2007, pp. 1424-1432
- [12] X. Zhang *et al.*, “DONet/CoolStreaming: A Data-Driven Overlay Network for P2P Live Media Streaming,” *IEEE INFOCOM*, vol. 3, Mar. 2005, pp. 2102–2111.
- [13] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, “Insights into pplive: A measurement study of a large-scale P2P IPTV system,” in *Proc. IPTV Workshop Conjunction WWW*, May 2006.
- [14] S. Xie, B. Li, G. Y. Keung, and X. Zhang, “Coolstreaming: Design, Theory, and Practice,” *IEEE TRANSACTIONS ON MULTIMEDIA*, vol. 9, NO. 8, DECEMBER 2007, pp. 1661-1671.
- [15] M. Zhang, L. Zhao, Y. Tang, J. Luo and S. Yang, “Large-Scale Live Media Streaming over Peer-to Peer Networks through Global Internet,” in *Proceedings of ACM Multimedia*, 2005.

[16] N. Magharei and R.Rejaie, “PRIME: Peer-to-Peer Receiver-driven Mesh-based Streaming,” in Proceedings of IEEE INFOCOM, 2007.

[17] OMNet++, URL: <http://omnetpp.org/>.

