Efficiency and Stability of Peer-to-Peer File Sharing Systems

Wei Qian Sang

A Thesis

In

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of Requirement
for the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

August 2007

Your file Votre référence
ISBN: 978-0-494-34721-8
Our file Notre référence
ISBN: 978-0-494-34721-8

ABSTRACT

Efficiency and Stability of Peer-to-Peer File Sharing Systems

Wei Qian Sang

In recent years, Peer-to-Peer (P2P) applications, from traditional file sharing to P2P live audio/video streaming, have become very popular. Among the P2P applications, P2P file sharing remains one of the most popular applications and its traffic has been dominating the Internet bandwidth for the past years. In this paper, I will mainly study the efficiency and stability of the P2P file sharing systems.

In the P2P file sharing networks, the upload bandwidth of each peer is a very important resource of the network. The efficient use of it will affect the system performance significantly. Motivated by this fact, a stochastic model for P2P file sharing networks is proposed and numerically solved to analyze how the performance of a P2P file sharing network is affected by different parameters such as the piece numbers of the file, the number of neighbours of a peer, and the seed departure rate etc. Based on this result, some useful guidelines are also provided on how to design an efficient P2P system.

Stability is another important issue in P2P file sharing systems. In this thesis, a simple fluid model is used to analyze the stability of BitTorrent-like P2P file sharing networks. The resulting fluid model is a switched linear system and it is proved that such a system is always globally stable. Numerical results based on extensive simulations are also provided to support the theoretical proof.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor, Dr Qiu, who gave me the opportunity to pursue my Master's degree. Without his trust, understanding, encouragement, patience, and instructions, it is hard to imagine that I can succeed in this research study.

I also thank the faculty and staff of Concordia University who provided me assistance. Furthermore, during these study years, learning together with other students has greatly enriched my education and life. I would like to thank all of them for their support and encouragement. Their suggestions have made the task of research enjoyable.

Last but not least, I would like to thank my wife and my parents. Their love and encouragement have made me overcome difficulties and complete this study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Description
P2P	Peer-to-Peer
HTTP	Hypertext Transfer Protocol
FTP	File Transfer Protocol
SMTP	Simple Mail Transfer Protocol
UDP	User Datagram Protocol
Snap	Structured overlay Networks Application Platform
FCFS	First-come-first-served
DHT	Distributed Hash Table
URL	Uniform Resource Locator

CHAPTER 1 INTRODUCTION

1.1 LITERATURE REVIEW

During the past decades, the Internet has developed very quickly. As a result, the Internet has become a very important resource of information and sometimes, it is called "The largest library ever created". Through Internet, people can share information that they are interested in. File sharing has been one of the important ways to share information. In the current Internet, most file sharing is done in the Peer-to-Peer (P2P) fashion.

1.1.1 P2P Model and Client-Server Model

P2P is a communication model in which each node (or peer) [5] has the symmetric capabilities and responsibilities [4]. Many new Internet applications have been designed based on the P2P model, such as Instant Messaging, File Sharing, Collaborative Community, IP Telephony, and High Performance Computing [10]. According to statistics from Ellacoya Networks in June 2007 [6], P2P users account for 37 percent of total Internet traffic, as shown in Figure 1.1 [7].

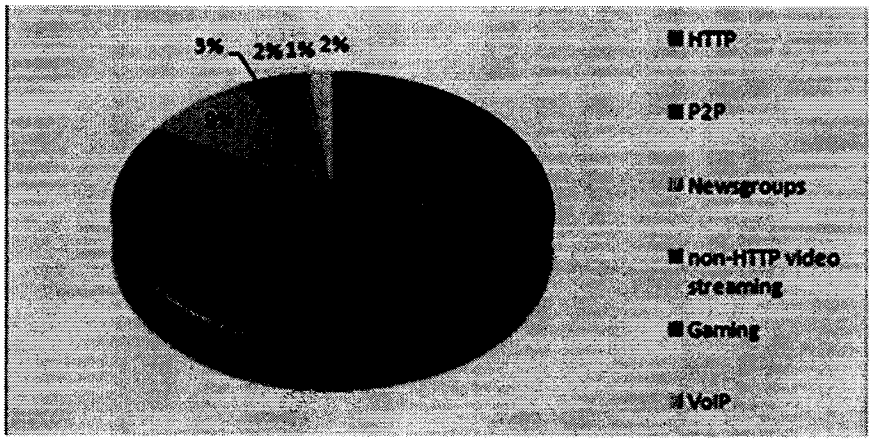

Figure 1.1: Percentages of total Internet traffic

The P2P model is significantly different from the client-server model. In the client-server model, each node plays an asymmetric role as either a client or a server and a server must be dedicated to serve other clients. On the other hand, in the P2P model, each communication node provides functionalities of both client and server. Both P2P model and client-server model have been widely used in Internet, and each has unique advantages and disadvantages.

The client-server model is the traditional model that has been adopted by many applications since the first day of the Internet. Client-server model is widly used in Internet protocols such as HTTP, FTP and SMTP [8]. When you access the Internet, no matter whether it is sending an email or just web browsing, you are in fact using the client-server model. Therefore, we can say that the client-server model is the foundation of the Internet.

With the centralization character, the client-server model has some advantages like ease of server maintenance, ease of updating resources, security and service assurance in dedicated servers etc., but the client-server model also has its own disadvantages. One of the main problems is the scalability. For example, the popular web service is built based on the client-server model. The web server normally works well if the load of service requests is light. However, for very popular web servers like google.com, cnn.com etc, there may be thousands of requests per second. Since a single server's capacity is limited, when the load is too high, the response time will increase significantly and users may have to wait for several minutes for a simple web page to be displayed. Currently, the solution is to use expensive servers with higher capacity and use multiple servers instead of a single server. However, the solution is not elegant and the scalability is still limited. The P2P model, on the other hand, could be a good solution to the scalability issue.

To overcome the poor scalability of the client-server model, the P2P model has been introduced and developed. As we discussed before, the difference between the P2P model and the client-server model is that with the P2P model, every node plays the same role as both client and server, while in the client-server model, every node has different role as either a client or a server. Generally, P2P systems are developed in a distributed fashion, which means that normally there is no single centralized server in the system. While on the other hand, a centralized server is normally necessary for client-server systems. The differences between Client-Server model and P2P model can be shown in Figure 1.2.

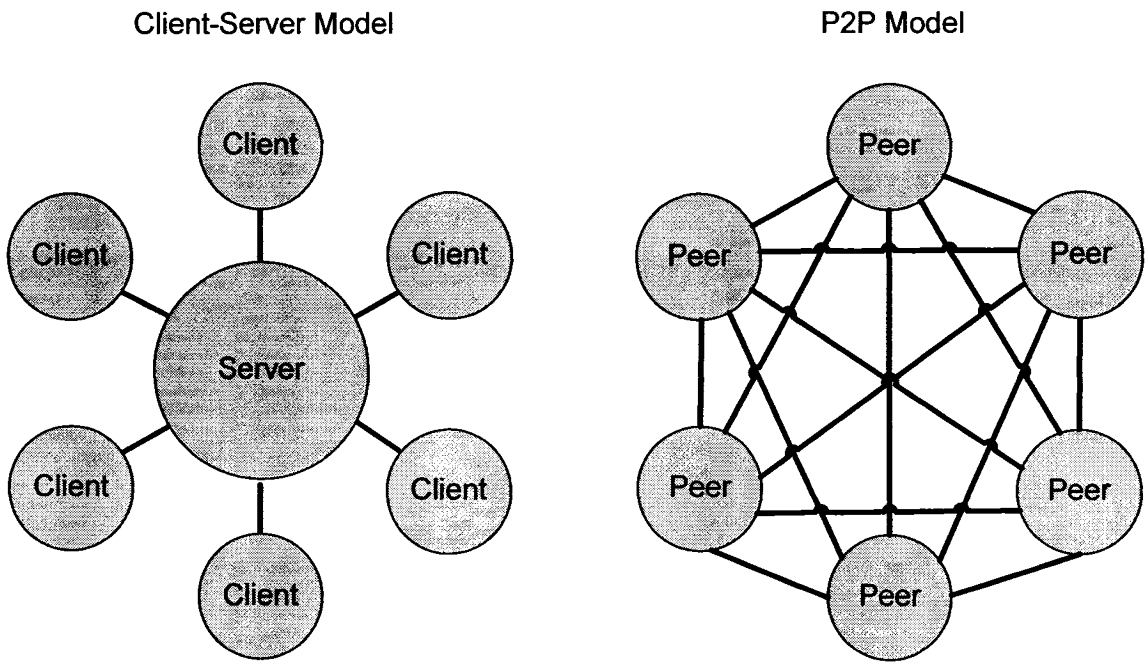

Figure 1.2: Client-server model and P2P model

However, the P2P model is not a replacement of the traditional client-server model. Instead, it should be looked upon as a good supplementary to the client-server networks. In some applications, the two kinds of models are combined together to provide better performance. For example, the Structured overlay Networks Application Platform (Snap) [9] has been developed to combine the P2P model with the traditional client-server web application model and provides significant performance improvement over the traditional client-server based web server as well as new features such as load balancing and fault - tolerance [9].

With the distributed and symmetric nature, the P2P model has advantages as follows:

Scalability

- In a P2P system, since each node plays as a client and a server at the same time, when the load of the system increases (which means there are more clients requesting service from the system), the number of servers also increase and hence the service provided by the system increases. From the user's point of view, the performance of the system will not degrade when there are more users joining the system. Hence, it has very good scalability.

Anonymity

- Peers (nodes) in P2P networks could be anonymous to each other.

Ease of sharing

- Peers can choose the best neighbours to communicate.
- Does not require a publication step (create a web page or upload to a server) to share information

Low cost

- No demand for high capacity and expensive servers.
- No special managements or administrative issues.

Performance

- Performance could be better when there are more peers.
- Good load balance and no bottlenecks in servers as in the client-server mode.

Fault Resilience (robustness and reliability)

- System breakdown due to single point failure can be avoided.
- Dependence on a central server can be reduced.

1.1.2 Introduction of P2P File Sharing Systems

In a P2P file sharing system, the certain files are stored on users in the network. These users are called peers or nodes and usually are individual computers. There is little or no centralized control among these peers. Each peer acts as both a client and a server at the same time. Each peer can initiate file requests to other peers, and also responses to file requests from other peers. Usually, the downloading and uploading activities are going on at the same time but neither of them is compulsory.

Before P2P file sharing became popular, the client-server model had been the most commonly used method for file sharing on the Internet. With the client-server model, file sharing can be easy to set up with http or ftp protocols. Even today, this method has still been widely used. However, as we discussed before, the client-server model doesn't work well when the load of the server is very heavy and hence has very poor scalability. For a given server, when the shared file is very large or there are too many client requests for the file, the limited capacity of the server will seriously affect the response time of the server and the download speed that the clients can obtain. This is the reason why P2P file sharing applications have become so popular during recent years.

In the first generation of P2P file sharing networks, the client-server model and the P2P model are mixed together. For example, in the Napster network [11,12,15,16], it is still necessary to have a central server, which is normally utilized as the centralized file index server. This kind of P2P model is called as centralized peer-to-peer model. In this model, when a peer joins the file sharing network, it will first communicate with the server for getting a list of peers that have the files. It can then start downloading the files from those peers. Similar with the client-server model, the centralized P2P networks have a single point of failure at the central server. Once the central server fails, the whole network may not work anymore.

After Napster encountered legal troubles, the decentralized P2P model was introduced into file sharing. New applications such as eDonkey [11,12,17,18], Gnutella [11,12,31], BitTorent [11,12,24,27,34], FastTrack [11,12,20,21] were developed. These applications are called the second generation and have soon become extremely popular. The traffic generated by these applications is dominating the current Internet bandwidth.

There are also two other generations of P2P applications that have been developed. The third generation is the indirect and encrypted P2P network [11, 12], and the fourth generation is the P2P networks that support media streams [11, 12]. The third generation of peer-to-peer networks are those that have anonymity features built in. And the fourth generation of peer-to-peer networks supply the users with services that can send streams instead of files over a P2P network, thus users can share live video without a centralized server involved [11,12].

1.1.3 Introduction of BitTorrent

In the past few years, traffic generated by P2P file sharing has dominated the Internet bandwidth. For example, in Germany, according to the P2P survey 2006 [26], the P2P file sharing is still on the rise. This survey was conducted during April to October 2006. In the daytime, P2P traffic accounts for 30% share of all Internet traffic and it is 70% in the nighttime. Between June and October 2006, the absolute data volume of P2P traffic has risen by 10%. Among all P2P applications, BitTorrent occupied more than half of all P2P traffic, which makes BitTorrent the most popular file sharing application in Germany. In this thesis, the efficiency and stability of BitTorrent-like P2P file sharing systems will be studied.

BitTorrent is a very popular P2P file sharing application. In a BitTorrent network, a peer can download the file of interest from other peers and at the same time, it can act as a server to upload to other peers. It uses a tit-for-tat [34] strategy as an incentive mechanism to encourage peers to upload. In real networks, people find that BitTorrent works extremely well for file sharing, especially for large, popular files. Many experiments have also shown that BitTorrent has very good scalability. When the network becomes very large, the performance of the network doesn't degrade as what has been observed with traditional HTTP or FTP protocols.

In the early versions of BitTorrent, special servers are still necessary for the network to work properly. To download files from a BitTorrent network, a peer should first obtain a torrent file. A torrent file is a small file that stores information about the file of interest, such as the file's length, name, hashing information, and the Uniform Resource Locator (URL) of a tracker. These torrent files are normally listed on some file sharing websites and there are also search engines that can be used to search the torrent files. Once a peer obtains the torrent file, it connects to the tracker specified in the torrent file. A BitTorrent tracker is an Internet server, which coordinates the communication between peers using the BitTorrent protocol. The most important function of the tracker is to return a list of peers that have the file of interest to the newly joined peer. As long as the new peer has the information of other peers, it can start downloading the file from those peers and the file exchanging process between peers will begin. During the file exchanging period, peers will periodically communicate with the tracker and update statistics to the tracker.

If the tracker goes down after a peer has already begun the file exchanging process with other peers, the peer can continue its file exchanging. But no new peers will be able to join the network anymore since they have no way to obtain information about other peers.

In BitTorrent file sharing networks, the tracker could be considered as the possible single point of failure. To improve the reliability of BitTorrent, multi-tracker torrents and tracker-less torrents are developed. With multi-tracker torrents, multiple trackers are used instead of a single tracker which improves the reliability in the case that if a tracker fails. With tracker-less torrents, every peer in the file sharing network can act as a tracker and hence eliminate the requirement of a single tracker server. There are mainly two kinds of tracker-less torrents methods: DHT-based implementations and Azureus 'Distributed Database' [24,34] .

In P2P file sharing, files are divided into pieces of fixed size. For BitTorrent networks, the default piece size is a quarter megabyte. Periodically, a peer will exchange information with other peers about which pieces has already been downloaded. By this method, a peer can know which pieces its neighbors have. Furthermore, pieces are further broken into sub-pieces, which are usually sixteen kilobytes in size, and typically five sub-piece requests are sent at once.

During the downloading process, peers usually have different pieces. When a new peer joins the network, it has no pieces at all and hence can't serve other peers. On the other hand, peers with most pieces of this certain file may receive many requests and hence will be busy on uploading. To select suitable pieces to request and to request them in a good order are important for good performance. In BitTorrent, there are four major piece selection strategies: strict priority, rarest first, random first and endgame mode. Strict priority means that once a peer requests a single sub-piece in one particular piece, the remaining sub-pieces of this piece will be requested before sub-pieces from any other piece for this peer. This strategy guarantees that the peer can get complete pieces quickly. Rarest first means that the rarest piece in the network should be downloaded first. It is designed to improve the survivability of the network. For example, if a piece has only one copy in the network and the peer that has this piece leaves the network, then no peers in the network can download the whole file anymore. With the rarest first policy, the possibility of this situation will be decreased. Random first is an exception to rarest first and it is applied only to new peers. When a peer is new and has no pieces at all, the peer will randomly select a piece to download. After the first piece is downloaded completely, the peer changes to rarest first policy. This strategy ensures that new peer can get the first piece as quickly as possible and can start to serve other peers soon. Endgame mode is applied when a peer has downloaded all pieces except the last one. In this situation, the peer will send the last piece request to all neighbor peers. This strategy is used to avoid the delay of the download. However it may also cause the peer to receive multiple copies of the last piece and hence wastes some bandwidth. The four strategies work together and make the piece selection algorithm of BitTorrent. [24, 34]

1.2 RELATED WORK AND RESEARCH OBJECTIVES

BitTorrent has been one of the most popular P2P file sharing applications and has attracted a lot of research attention. While early work on P2P systems has mainly focused on system design and traffic measurements, [27, 28, 41], some recent research has emphasized performance modeling. In [29], a closed queuing system is used to model a general P2P file sharing system and basic insights on the stationary performance are provided. In [30, 31], a stochastic fluid model is used to study the performance of P2P web cache (SQUIRREL) and cache clusters. In [36, 38], a branching process is used to study the service capacity of BitTorrent-like P2P file sharing in the transient regime and a simple Markovian model is presented to study the steady-state properties. In [46], a spatio-temporal model is proposed to analyze the resource usage of P2P systems. In [47], an approximation for the life time of a chunk in BitTorrent is proposed. In [48], the authors present an extensive trace analysis and modeling study of BitTorrent-like systems. In [42], the authors studied the behavior of peers in BitTorrent and also investigated the file availability and the dying-out process. In [37], a simple fluid model is proposed to study the performance and scalability of BitTorrent-like P2P systems.

Although a lot of research has been conducted in the area of P2P file sharing, there are still some important issues that are not understood very well. In this thesis, two important problems in P2P file sharing, the efficiency and the stability will be studied. More specifically, the objectives of this research are shown as follows:

Propose a stochastic model to study the utilization of upload bandwidth in BitTorrent-like P2P file sharing systems.

By solving the proposed stochastic model, gain some insights on how the performance of a P2P file sharing network is affected by different parameters.

Obtain some useful guidelines on how to design an efficient P2P file sharing system.

Study the stability of the fluid model proposed in [37]. More specifically, prove that the fluid model proposed in [37] is globally stable.

Simulate BitTorrent-like P2P file sharing systems extensively and verify the theoretical results that have been obtained.

1.3 THESIS ORGANIZATION

This thesis will be presented as follows:

Chapter 2 Efficiency of P2P file Sharing: In this chapter, the efficiency problem of P2P file sharing system is analyzed for a BitTorrent-like system. In section 2.1, the motivation is described. In section 2.2, assumptions of the system are given. Under the condition that the system is in steady state, a stochastic discrete-time model is proposed and analyzed. In section 2.3, by numerically solving the model proposed in section 2.2, we obtain some important insights on how the performance of P2P file sharing will be affected by different parameters. Then in the last section, we will give guidelines on how to design an efficient P2P file sharing system based on our theoretical and numerical results.

Chapter 3 Analysis of Global Stability: In this chapter, the global stability of BitTorrent-like P2P file sharing systems is analyzed in detail. In section 3.1, the motivation of this study is described. In section 3.2, the simple fluid model proposed in [37] is discussed. In section 3.3, the local stability of the fluid model is summarized. In section 3.4, the global stability of the fluid model is analyzed under three different initial conditions. Under all these initial conditions, the global stability of the system is proved. In section 3.5, the fluid model is numerically simulated under different initial conditions to verify our theoretical results. In the last section, the contribution of this study is summarized.

Chapter 4 Conclusions and future Work: In section 4.1, the conclusions of this thesis are summarized. In section 4.2, some possible future works are discussed.

CHAPTER 2 EFFICIENCY OF P2P FILE SHARING

2.1 MOTIVATION

As has been discussed in Chapter 1, in P2P file sharing, a certain file is divided into many pieces. The size of each piece ranges from several hundred kilobytes to several megabytes. When a new peer joins the network, it begins to download pieces from other peers. As long as it obtains one piece of the file, the new peer can start to serve other peers by uploading pieces. Since peers are downloading and uploading at the same time, when the network becomes large, although the demands increase, the service provided by the network also increases. Hence, the performance of the P2P network scales very well.

In a P2P file sharing network, each peer contributes to the network through uploading and hence the upload bandwidth of each peer is a very important resource of the network. The efficient use of it will impact the system performance significantly. However, little research has been done in this area. We first observe that the number of pieces that a given peer has is an important factor that affects the upload bandwidth utilization. For example, when a peer first enters the network, it has no pieces at all and hence can not upload to anyone. The upload bandwidth utilization is zero in this case. On the other hand, when a peer has most of the pieces, it is very likely that it can upload to others and hence the utilization is close to one. Motivated by this fact, a stochastic model is proposed to study the peer distribution with regards to the number of pieces that a peer has. More specifically, P_i will be determined, which is the probability that a random peer has i pieces, where $0 \le i \le N$ and N is the total number of pieces of the served file. Note that in BitTorrent, peers that have the whole file are called seeds, while other peers are called downloaders. After the peer distribution is obtained, this result is used to study how the efficiency is related to different network parameters, such as the number of pieces, the number of neighbours, and the seed departure rate.

2.2 STOCHASTIC MODEL

2.2.1 Model Assumptions

In this section, the assumptions for the stochastic model will be described:

BitTorrent-like P2P file sharing network

There are many P2P file sharing applications available in the current Internet. However, BitTorrent has been one of the most popular applications for the past years. So, in this thesis, BitTorrent-like P2P file sharing networks are studied.

Steady State

From real trace measurements, it has been observed that three are three stages in BitTorrent-like P2P file sharing networks, a growing stage, a stabilizing stage, and a decaying stage. The stabilizing phase is normally the one that most of downloads take place and hence it is the one that determines the performance of the system. In this chapter, a P2P file sharing system that is already in the steady state is considered.

Piece number N

In P2P file sharing systems, a certain file is divided into many pieces. In this chapter, the total number of pieces of the served file is assumed to be N.

Peer distribution P_i

Peer distribution Pi indicates the probability that a random peer has i pieces, where $0 \le i \le N$. In this chapter, the P2P network is assumed to be very large and in the steady state. Hence the peer distribution $\{P_i\}$ doesn't change with time.

Number of Neighbours L

Peers in a P2P network will communicate with each other. When a peer enters the network, it first gets a list of peers from the tracker. These peers are called the new peer's neighbours. In this chapter, for the simplicity of analysis, I assume that the number of neighbours of each peer is fixed at L.

Piece distribution

For a given peer with i pieces, I assume that these pieces are chosen randomly from the set of all pieces of the file. This is a reasonable assumption because BitTorrent-like systems take a rarest first piece selection policy when downloading. Hence, it is unlikely that the network has significantly more copies of a piece than copies of other pieces. I also assume that these pieces are chosen independently with other peers. In P2P networks, a peer is normally downloading from many neighbor peers at the same time. Hence, a single neighbor's effect on the given peer's pieces can be neglected and it is reasonable to assume the independence between peers.

Upload and Download bandwidth

Since we are interested in the efficient use of upload bandwidth, for simplicity of analysis, I assume all peers have the same upload bandwidth and the download bandwidth is unlimited. I use a discrete time model and without loss of generality, I assume that the upload bandwidth of a peer is one piece per time slot.

No downloader departure

For the simplicity of analysis, I assume that once a peer enters the network, it will not leave the network until it downloads the whole file. Hence, there is no downloader departure in this model.

Seed departure rate γ

The seed departure rate γ represents the probability that a seed leaves the system in any given time slot.

Peer arrival rate λ

The peer arrival rate λ represents the number of new peer arrivals per time slot.

Other notations that will be used in this chapter are also listed below:

$F(i, j)$ probability that peer B has no pieces that peer A is interested in, where peer A has i pieces and peer B is a neighbour of peer A with j pieces.

$r_{i,k}$ the probability that a peer with i pieces downloads k pieces in the current time slot.

M the average number of peers in the system.

T the average time a peer stays in the system.

T_d the average download time of a peer. (i.e., the time from the moment that a peer enters the system until it becomes a seed)

T_N the average time a seed stays in the system.

Note that although our model is relatively simple, it has all the important features of a typical P2P file sharing network and we expect the model can shed some light on further research of P2P file sharing efficiency.

2.2.2 Model Analysis

With the assumptions described in last section, the stochastic model can be shown as Figure 2.1.

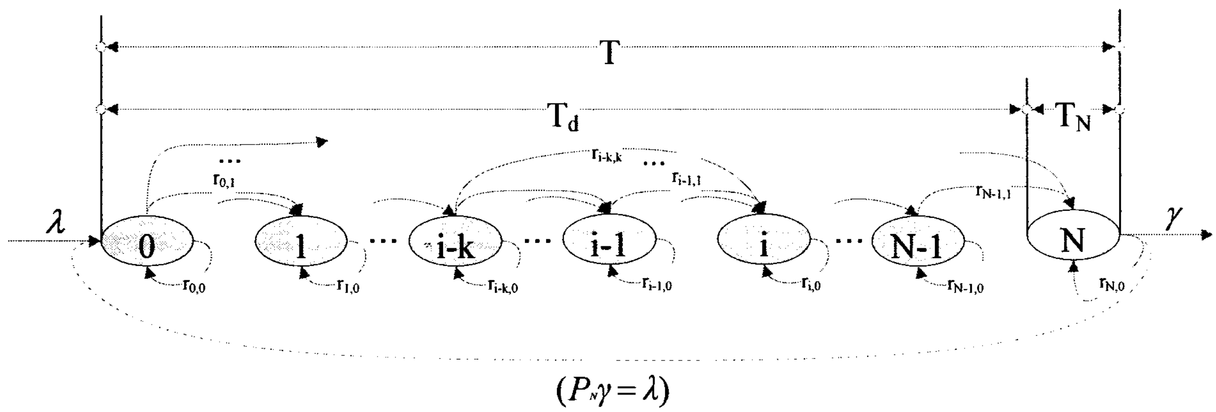

Figure 2.1: A stochastic and discrete-time BitTorrent-like P2P file sharing model

In BitTorrent-like systems, when a peer obtains a new piece, it will update this information with its neighbours and hence a peer knows what pieces its neighbours have. At the beginning of a time slot, a peer will send requests for pieces to its neighbours if a neighbour has pieces that the peer is interested in. At the same time, the peer will also receive requests from its neighbours. If the peer receives more than one request, it will randomly pick up one of the requests to fulfill. Note that in a real BitTorrent network, peers will fulfill requests according to a built-in incentive mechanism. How the incentive mechanism will affect the performance is out of the scope of this thesis and is part of the future work.

Next, the analysis of two peers in a neighbourhood will be considered. It is assumed that peer A has i pieces and peer B is a neighbour of peer A with j pieces. $F(i, j)$ is the probability that peer B has no pieces that peer A is interested in, i.e., peer A has all the pieces that peer B has, then $F(i, j)$ can be expressed by,

$$F(i,j) = \begin{cases} 0, & i < j \\ \dfrac{\binom{N-j}{i-j}}{\binom{N}{i}}, & i \geq j \end{cases} \tag{1}$$

In a given time slot, if peer B has pieces that peer A is interested in, peer A will send a piece request to peer B. Since peer B has L neighbour peers, it may receive more than one piece request. According to our assumptions, if peer B receives more than one request, it will randomly pick up one request to fulfill and the piece request from peer A will only have a certain probability to be fulfilled. Next, the probability that peer A could get one piece from peer B in the given time slot will be calculated.

Assuming X be the number of requests that B receives besides A's request, then X is a Binomial random variable with parameters $L-1$ and q_j. Here $L-1$ is the maximum number of requests B could receive besides A's request; q_j is the probability that a randomly picked neighbour of B sends request to B. q_j can be expressed as follows,

$$q_j = \sum_{k=0}^{N} P_k (1 - F(k,j)) \tag{2}$$

Let Prob$\{X=k\}$ be the probability that $X=k$, which means that besides peer A's request, the number of requests that B receives is equal to k. Because X is a Binomial random variable, we have

$$Prob\{X=k\} = \binom{L-1}{k} q_j^k (1-q_j)^{L-1-k} \tag{3}$$

The probability that peer B fulfill the request of peer A will then be,

$$\begin{aligned}
G_j &= \sum_{k=0}^{L-1} \frac{1}{k+1} Prob\{X=k\} \\
&= \sum_{k=0}^{L-1} \frac{1}{k+1} \binom{L-1}{k} q_j^k (1-q_j)^{L-1-k} \\
&= \sum_{k=0}^{L-1} \frac{(L-1)!}{(k+1)!(L-1-k)!} q_j^k (1-q_j)^{L-1-k} \\
&= \frac{1}{Lq_j} \sum_{k=0}^{L-1} \frac{L!}{(k+1)!(L-1-k)!} q_j^{k+1} (1-q_j)^{L-1-k} \\
&= \frac{1}{Lq_j} \sum_{k=0}^{L-1} \binom{L}{k+1} q_j^{k+1} (1-q_j)^{L-1-k} \\
&= \frac{1}{Lq_j} \sum_{m=1}^{L} \binom{L}{m} q_j^m (1-q_j)^{L-m} \\
&= \frac{1-(1-q_j)^L}{Lq_j}
\end{aligned} \tag{4}$$

With Eq. (4), we can obtain the probability that peer A downloads a piece from a randomly selected neighbour peer,

$$S_i = \sum_{j=0}^{N} P_j (1 - F(i,j)) G_j \,. \tag{5}$$

According to the BitTorrent protocol, at any given time slot, a peer only sends one request for a given piece. Therefore, for a given peer with i pieces, the maximum number of requests that can be sent by this peer is $D=min(L, N-i)$. We can see that the number of pieces that it downloads in the same time slot is then a Binomial random variable with parameters D and S_i. The probability that a peer with i pieces downloads k pieces in this time slot, can be expressed as

$$r_{i,k} = \binom{D}{k} S_i^k (1-S_i)^{D-k}, \tag{6}$$

where i=0, 1, … , N-1 and k=0, … , D.

With Eq. (6), we can obtain the average download rate of a peer with i pieces,

$$d_i = \sum_{k=1}^{\min(L,N-i)} k r_{i,k} \tag{7}$$

Since the system is in the steady state, the peer distribution doesn't change with time. So we have

$$P_i = \sum_{k=0}^{\min(i,L)} P_{i-k} r_{i-k,k} \tag{8}$$

where i=1, 0, … , N.

When a peer has downloaded all the pieces, it becomes a seed. Once a peer becomes a seed, it will serve other peers until it leaves this system. In the last section, we have defined the seed departure rate as γ and peer arrival rate as λ. Since the system is in a steady-state, the total seed departure rate equals to the new peer arrival rate. Hence, we have

$$P_0 = P_0 r_{0,0} + P_N \gamma . \tag{9}$$

Solving Eq. (6), Eq. (8) and Eq. (9), we can obtain the peer distribution $\{P_i\}$, where i=0, ... , N. Note that it is hard to get a closed form peer distribution. However, the equations can easily be numerically solved and the results will be discussed in the next section.

Once we know the peer distribution, we can use it to study important performance such as the average download time of a P2P network. Let M denote the average number of peers in the system, T denote the average time a peer stays in the system, and T_d denote the average download time of a peer. Applying the Little's Law [33, 39] to the whole system, the seeds, and the downloaders respectively, we can get the following equations relating to M, T, and T_d,

$$\begin{cases} M = \lambda T \\ MP_N = \lambda \dfrac{1}{\gamma} \\ M(1-P_N) = \lambda T_d \end{cases} \tag{10}$$

From Eq. (10), it is easy to get the average time a peer stays in the system:

$$T = \frac{1}{P_N \gamma} \tag{11}$$

and the average downloading time:

$$T_d = \frac{1-P_N}{P_N \gamma} \tag{12}$$

Next, we will study how the system performance can be affected by different parameters such as the number of pieces N, the number of neighbours L, and the seed departure rate γ etc.

2.3 NUMERICAL RESULTS

Peer Distribution

This simulation is under the condition that the total number of pieces N=200, the number of neighbours L=20, and the seed departure γ=0.01. Figure 2.2 presents the peer distribution as a function of the number of pieces that a peer has.

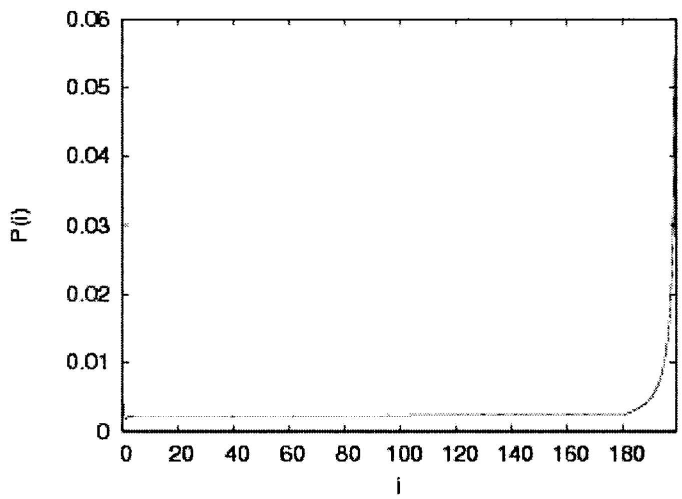

Figure 2.2: Peer distribution

We can see when i<180, peers are almost uniformly distributed. And when i>180, the probability that a peer has i pieces increases when i increases. That is because when one piece has most pieces (i>180), it can't fully utilize all of its neighbours anymore and hence its download rate decreases. We call it the end-game effect. In BitTorrent, there is an end-game mode to deal with this issue.

Download Rate Distribution

This simulation is under the condition that the total piece number N=200, neighbour number L=20, and the seed departure γ=0.01. Figure 2.3 presents the average download rate as a function of the number of pieces that a peer has.

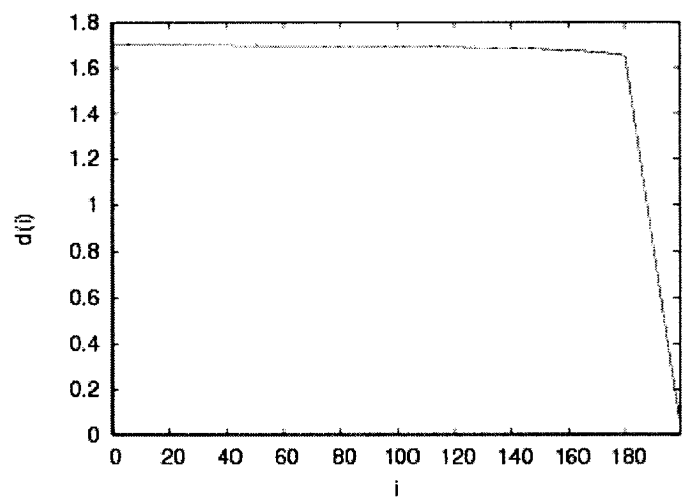

Figure 2.3: Download rate distribution

We can clearly see that when i<180, the download rate is almost a constant. This explains the reason that in BitTorrent, the download rate of a peer can normally remains at a high value for most part of the download process. When i>180, however, the download rate keeps decreasing and that also explains the result in Figure 2.2.

The Effect of Seed Departure Rate

This simulation is under the condition that the total piece number N=200, neighbour number L=20, and the seed departure γ is changed from 0.01 to 0.9. Figure 2.4 presents the normalized download time $\frac{T_d}{N}$ as a function of the seed departure rate γ.

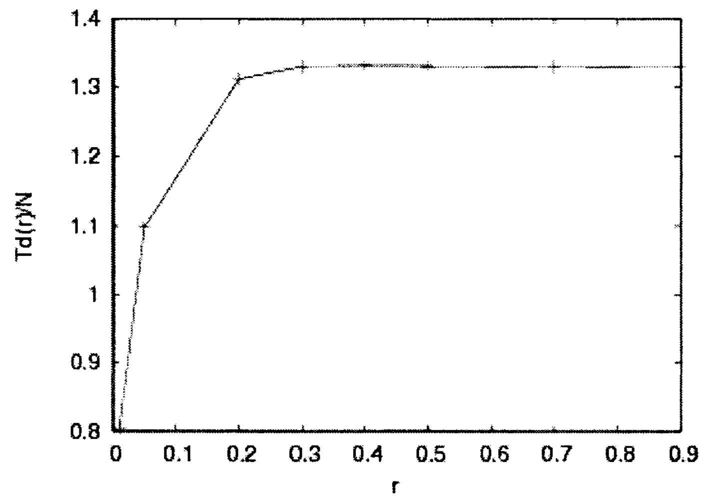

Figure 2.4: The effect of seed departure rate

We can see when γ increases, the download time also increases because there is less seeds in the system. However, if γ>0.3, when γ increases, the download time doesn't change much. This tells us that the seed departure rate affects the system performance but it is not significant once γ is greater than some thresholds. Note that when γ is too large, it may happen that no single seed in the system and hence cause the survivability problem of the network.

The Effect of Piece Number

This simulation is under the condition that neighbour number $L=20$, the total piece number N and the seed departure γ satisfying $\gamma N=2$, and N from 25 to 300. It shows that how the average download time is affected by the number of pieces.

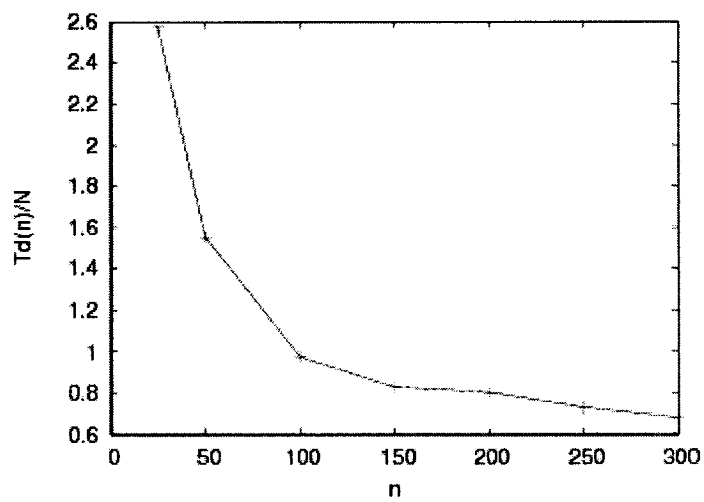

Figure 2.5: The effect of piece number

In Figure 2.5, we can see that when N increases, the average download time decreases as expected since larger N means a peer is more likely to upload to its neighbours. And with the increase of piece number, the decrease of average download time is initially very sharp, and then slows down. Note that we are keeping γN a constant because when N increases, the length of each time slot decreases (assuming the file size is a constant), hence we need to adjust γ accordingly.

The Effect of Neighbour Number

This simulation is under the condition that N=200, the seed departure γ=0.01, and the neighbour number L from 2 to 20. It shows that how the average download time is affected by the number of neighbours.

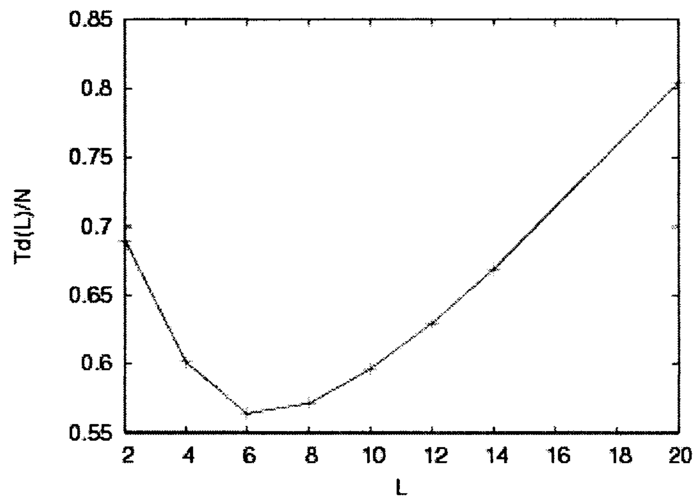

Figure 2.6: The effect of neighbour number

In Figure 2.6, we see that when L=6, the download time is the smallest. The explanation is that when L is too small, the probability that a peer can upload to its neighbours is small and hence not very efficient. However, when L is too large, the end game effect will be significant and will increase the download time.

2.4 CONCLUSION

In this chapter, a stochastic model has been proposed to analyze the efficiency of P2P file sharing. By solving the model numerically, we are able to gain some important insights on how the performance of P2P file sharing is affected by different parameters. Furthermore, based on these results, some guidelines can be obtained to design an efficient P2P file sharing system.

The following conclusions are drawn from the results:

1. The end game stage affects the performance significantly. To improve the performance, it is important to alleviate the end game effect.

2. If not considering the survivability, the seed departure rate doesn't affect the performance significantly when it is large enough. Hence, as long as we have at least one seed in the system, it is not necessary to ask seeds to stay in the system for a long period of time.

3. Too many neighbours may affect the performance adversely since it causes more severe end game effect. Therefore, it is important to choose a reasonable number of neighbours.

CHAPTER 3 ANALYSIS OF GLOBAL STABILITY

3.1 MOTIVATION

From real trace measurements, it has been observed that a BitTorrent-like P2P file sharing network has three phases [35], a growing phase, a stabilizing phase, and a decaying phase. In the stabilizing phase, the system enters a "steady state", in which the number of peers and the performance of each peer are relatively stable. The stabilizing phase is normally the one that most of the downloads take place and hence it has significant impact on the system performance. A lot of research has been done on the performance analysis in the stabilizing phase. However, theoretically, it is still not clear to us whether a P2P file sharing system always has a stabilizing phase. And if not, what are the conditions required for the system to be stable. For example, if the number of peers in the system never enters a steady state and keeps oscillating, it will be meaningless to analyze the steady state performance. Hence, the stability of a P2P system is a fundamental problem for any serious P2P performance analysis which is addressed in this chapter. As far as we know, this is the first work to theoretically analyze the global stability of BitTorrent-like P2P networks. Note that there are some works in the literature on the stability of P2P networks. However, the stability studied in those works is not as same as the stability we studied here. For example, in [44], the stability of Chord-based P2P systems is studied. The stability there, however, is mainly about how stable or robust the overlay network connections can be maintained if there are peers leaving the system. While in this chapter, we are interested in whether the system will ever enter into a steady state, in which we can then analyze the performance such as the average download rate of each peer etc.

3.2 FLUID MODEL

In this chapter, the simple fluid model proposed in [37] will be used to analyze the global stability of BitTorrent-like systems. In [37], the performance and the local stability of BitTorrent-like P2P systems have been studied, and some useful results have been obtained. In this section, a description of the fluid model will be given firstly.

In BitTorrent systems, there are two types of peers. The first kind of peers is seed, which has all pieces of the served file and only performs uploading. The second kind of peers is downloader, which has only partial (or none) of the file and can perform downloading and uploading at the same time. Without loss of generality, we assume the size of the served file is 1, and the following notations will be used to describe a BitTorrent-like P2P network that serves the given file [37]:

$x(t)$ number of downloaders (also known as leechers) in the system at time t.

$y(t)$ number of seeds in the system at time t.

λ the arrival rate of new peers and $\lambda>0$.

μ the uploading bandwidth of a given peer and $\mu>0$. We assume that all peers have the same uploading bandwidth.

c the downloading bandwidth of a given peer and $c \geq \mu$. We assume that all peers have the same downloading bandwidth.

θ the rate at which downloaders abort the download and $\theta \geq 0$.

γ the rate at which seeds leave the system and γ >0.

η indicates the effectiveness of the file sharing, and η takes values in [0, 1]. More details about η can be found in [37, 38].

Here we assume that the seed departure rate γ >0, which is always true in real networks since no peer will stay in the system forever. Note that in the unrealistic case γ =0, seeds never leave the system. Hence eventually the number of seeds in the system will go to infinity and the system is unstable. In this chapter, we always assume γ >0.

The deterministic fluid model for the evolution of the number of peers (downloaders and seeds) used in [37] is given by:

$$\frac{dx}{dt} = \lambda - \theta x(t) - \min\{cx(t), \mu(\eta x(t) + y(t))\},$$
$$\frac{dy}{dt} = \min\{cx(t), \mu(\eta x(t) + y(t))\} - \gamma y(t). \tag{12}$$

The minimum operation in Eq. (12) determines whether the uploading bandwidth μ or the downloading bandwidth c is the constraint of the system. Because of the minimum operation, the whole system is a non-linear system.

Before we start the stability analysis, we will give some stability definitions [32] firstly.

For a system $\dot{X} = f(x)$, where $f: R^n \rightarrow R^n$ may be linear or non-linear, a point x_e is an equilibrium point if $f(x_e) = 0$. The system is globally stable if for every trajectory $x(t)$, we have $x(t) \rightarrow x_e$ as $t \rightarrow \infty$. The system is locally stable near x_e if there is an $R > 0$, such that if the initial condition $x(0)$ is near the equilibrium point, i.e., $||x(0) - x_e|| \leq R$, then $x(t) \rightarrow x_e$ as $t \rightarrow \infty$. Obviously, if a system is globally stable, then it must be also locally stable. The reverse is generally not true. However, if the system is linear, i.e., $f(x)=Ax$, then local stability is equivalent to global stability. A linear system is stable if and only if all the eigenvalues of the matrix A have negative real parts.

Letting $\frac{dx}{dt} = \frac{dy}{dt} = 0$ in Eq. (12), we can obtain the equilibrium point of the system as

$$\bar{x} = \frac{\lambda}{\beta(1+\frac{\theta}{\beta})}$$

$$\bar{y} = \frac{\lambda}{\gamma(1+\frac{\theta}{\beta})} \tag{13}$$

where $\bar{x}$ and $\bar{y}$ are the equilibrium values of $x(t)$ and $y(t)$ respectively and $\frac{1}{\beta} = \max\left\{\frac{1}{c}, \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})\right\}$. However, it is still not clear to us whether this equilibrium point is stable or not. As we will see later in next section, the system is governed by two different linear systems depending on whether $cx > \mu(\eta x + y)$ or not. This type of system is called switched linear system and it is normally hard to determine the global stability of such a system.

Next, we first summarize the result of local stability of the system and discuss the difficulties in determining the global stability.

3.3 LOCAL STABILITY

From Eq. (12), we see that the system behaves differently when $cx>\mu(\eta x+y)$ and $cx<\mu(\eta x+y)$. Solving the equation $cx=\mu(\eta x+y)$ gives us $y=\frac{c-\mu\eta}{\mu}x$. For convenience, we define $k=\frac{c-\mu\eta}{\mu}$. When $y\geq kx$, $cx(t)\leq\mu(\eta x(t)+y(t))$ and the downloading bandwidth is the constraint. When $y<kx$, $cx(t)>\mu(\eta x(t)+y(t))$ and the uploading bandwidth is the constraint. So $y=kx$ divides the (x, y) plane into two areas (see Figure 3.1).

Note that the equilibrium point obtained in Eq. (13) may fall into area I or area II depending on the system parameters. More specifically, when $\frac{1}{c}\geq\frac{1}{\eta}(\frac{1}{\mu}-\frac{1}{\gamma})$, we will have $\bar{y}\geq k\bar{x}$, which means that the equilibrium point is in area I and when $\frac{1}{c}<\frac{1}{\eta}(\frac{1}{\mu}-\frac{1}{\gamma})$, we have $\bar{y}<k\bar{x}$ and the equilibrium point falls into area II. When $\frac{1}{c}=\frac{1}{\eta}(\frac{1}{\mu}-\frac{1}{\gamma})$, the equilibrium point falls exactly on the line $y=kx$.

In area I ($y\geq kx$), we have

$$\frac{dx}{dt} = \lambda - \theta x(t) - cx(t),$$
$$\frac{dy}{dt} = cx(t) - \gamma y(t). \tag{14}$$

In area II ($y \le kx$), we have

$$\frac{dx}{dt} = \lambda - \theta x(t) - \mu(\eta x(t) + y(t)) = \lambda + (-\theta - \mu\eta)x(t) + y(t),$$
$$\frac{dy}{dt} = \mu(\eta x(t) + y(t)) - \gamma y(t) = \mu\eta x(t) + (\mu - \gamma)y(t). \tag{15}$$

Both Eq. (14) and Eq. (15) are linear systems. The whole system Eq. (12) is a so-called switched linear system.

The linear system Eq. (14) has two real eigenvalues $\lambda_1 = -(\theta + c)$ and $\lambda_2 = -\gamma$. Both eigenvalues are negative. So Eq. (14) is always a stable system. If the equilibrium point is in area I, it is then locally stable. Solving Eq. (14), we see that $x(t)$ and $y(t)$ take the following forms

$$x(t) = \bar{x} + x_1 e^{-(\theta+c)t}$$
$$y(t) = \bar{y} + y_1 e^{-(\theta+c)t} + y_2 e^{-\gamma t} \qquad \text{if } \gamma \ne \theta + c$$
$$y(t) = \bar{y} + y_1 e^{-\gamma t} + y_2 t e^{-\gamma t} \qquad \text{if } \gamma = \theta + c \tag{16}$$

where $(\bar{x}, \bar{y})$ is the equilibrium point and x_1, y_1, y_2 are some constants.

Note that in Figure. 3.1, we assume that $c \ge \mu\eta$ and hence $k \ge 0$. This is normally the case in real networks since $\eta \le 1$ and the downloading bandwidth c is normally greater than the uploading bandwidth μ. If $c < \mu\eta$, then we always have $cx < \mu(\eta x + y)$ and the whole system is always governed by Eq. (14). The system is reduced to a simple linear system and from the analysis above, the system is always stable (both locally and globally). Hence, in the rest of the paper, without loss of generality, we assume that $c \geq \mu\eta$.

In area II, the linear system described by Eq. (15) may or may not be stable depending on the parameters. For example, if $\gamma < \mu$, we see that when *t* goes to infinity, *y(t)* will also goes to infinity and hence the system is unstable. On the other hand, when $\gamma > \mu$, we can show that both eigenvalues of Eq. (15) have negative real parts. Hence the system is stable.

When the equilibrium point falls into area II, we have $\frac{1}{c} < \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})$. Hence $\gamma > \mu$ and the equilibrium point is locally stable.

In summary, no matter whether the equilibrium point is in area I or in area II, it is always locally stable. The only exception is when the system equilibrium point is on $y=kx$, the boundary of these two linear systems. In this case, we will not be able to use linear analysis to determine even the local stability of the system. Hence, simply studying the two linear systems separately is not enough and we need to consider the whole system together and study it's global stability.

Note that the system we studied here is a switched linear system. While the linear system in Eq. (14) is always stable, the other linear system in Eq. (15) may or may not be stable. Hence, depending on the system settings, the whole system may switch between two stable linear systems or between one stable and one unstable system. The global stability of such a system is normally very tricky and a survey on this issue can be found in [45].

The main difficulty in the stability analysis of a switched linear system lies in the fact that the system may be governed by two different linear equations alternatively. Even if both linear systems are stable, the whole system may still be unstable. Fortunately, for the fluid model in Eq. (12), we are able to apply the Lyapunov function [32] to prove that this system is indeed globally stable.

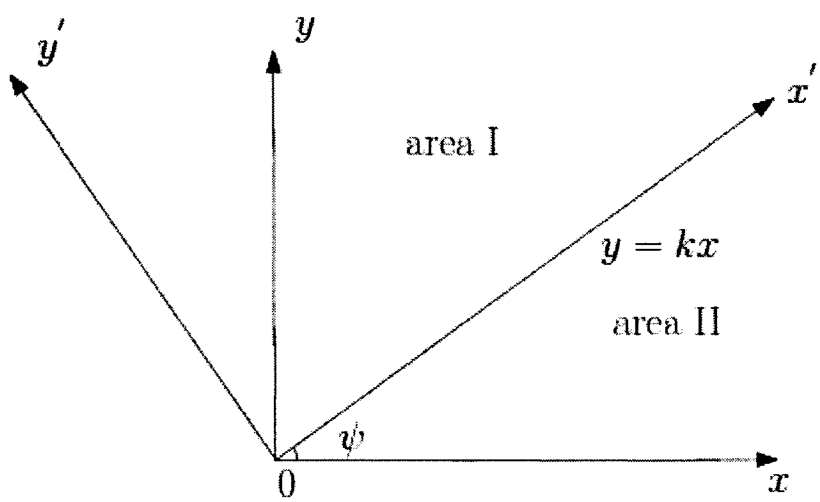

Figure 3.1: A linear transform of the system

3.4 GLOBAL STABILITY

We first prove a simple but useful property of the system described by Eq. (12).

Proposition 1 *If the initial condition of the system* $x(0) \geq 0$ *and* $y(0) \geq 0$, *then* $x(t) \geq 0$, $y(t) \geq 0$ *for all t.*

Proof: From Eq. (12), it is easy to see that if $x(t) = 0$,

$$\frac{dx(t)}{dt} = \lambda > 0,$$

and when $y(t) = 0$,

$$\frac{dy(t)}{dt} = u\eta x(t) \geq 0.$$

So $x(t) \geq 0$, $y(t) \geq 0$ if $x(0) \geq 0$ and $y(0) \geq 0$.

This proposition tells us that under any initial condition $x(t) \geq 0$ and $y(t) \geq 0$, the system will always be limited to the first quadrant of the (x, y) plane. For practical networks, this obviously should be true, since the number of seeds and downloaders cannot be negative. Although the proposition itself is simple and quite obvious, later we will see that the simple approach used here can be very useful in the study of the global stability of the fluid model in Eq. (12). Briefly speaking, we will study the global stability by applying a similar approach to that used in the proof of Proposition 1. What we will show is that no matter what the initial condition is (as long as $x(0) \geq 0$ and $y(0) \geq 0$), the system described by Eq. (12) will stay in either area I or area II forever after a finite time. Hence, after a finite time, the system is reduced to a simple linear system and we can then study its stability by using linear analysis.

For the convenience of analysis, we next do a linear transform of the system as shown in Fig. 3.1, where the x' axis is the switching line $y=kx$. Let $\Psi = \tan^{-1} k$. We then have

$$x' = x\cos\Psi + y\sin\Psi$$

$$y' = -x\sin\Psi + y\cos\Psi .$$

In the new (x', y') plane, if the system is in area I, $y' \geq 0$ and if it is in area II, $y' \leq 0$.

We are interested in how the system switches between area I and area II. So we will study $\frac{dy'}{dt}$ when $y' = 0$ (i.e., along the x' axis, or along the line $y = kx$). Along the x' axis, we have

$$\begin{aligned}
\frac{dy'}{dt} &= -\frac{dx}{dt}\sin\Psi + \frac{dy}{dt}\cos\Psi \\
&= -(\lambda - \theta x - cx)\sin\Psi + (cx - \gamma y)\cos\Psi \\
&= -(\lambda - \theta x - cx)\sin\Psi + (cx - \gamma kx)\frac{\sin\Psi}{k} \\
&= ((\frac{c}{k} + \theta + c - \gamma)x - \lambda)\sin\Psi .
\end{aligned}$$

Obviously, when the system enters area I from area II, we have $\frac{dy'}{dt} > 0$ and hence

$x > \dfrac{k\lambda}{c+k(\theta+c-\gamma)}$. Similarly, if the system enters area II from area I, $\dfrac{dy'}{dt}$<0 and we have $x < \dfrac{k\lambda}{c+k(\theta+c-\gamma)}$. Here we see that when the system switches between area I and area II, the number of downloaders *x(t)* has to satisfy certain conditions.

Next, we consider three cases: the equilibrium point in area I, in area II, and on the line *y=kx*. We will show that in all three cases, the number of occurrences of switching between area I and area II is always finite and hence after a finite time, the system will either stay in area I forever or stay in area II forever.

Lemma 1 If $\frac{1}{c} > \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})$, the system will stay in area I forever after a finite time and system is globally stable.

Proof: When $\frac{1}{c} > \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})$, the downloading bandwidth is the bottleneck and the equilibrium point of the system is in area I. If initially, *(x(0), y(0))* is in area I and *(x(t),y(t))* is in area I for all time t, then the system is obviously stable. If initially, *(x(0),y(0))* is in area II or the system enters area II after a finite time, we next prove that it will enter area I within a finite time. We consider two cases here, Eq. (15) is stable and Eq. (15) is unstable.

If $\gamma \geq \mu$, Eq. (15) is stable. But its equilibrium point is in area I. So after a finite time, the system will enter area I.

If $\gamma < \mu$, Eq. (15) is unstable. It is easy to show that with Eq. (15), $y(t)$ increases to 1 while $x(t)$ goes to zero. Again, after a finite time, the system will enter area I.

We denote the time that the system enter area I by t_0. Next, we will prove that when $t \geq t_0$, the system is always in area I.

Now, we will consider when the system is in (x', y') plane. After the system enters area I, Eq. (14) takes effect. If $\gamma \neq \theta + c$, from Eq. (16), we can write $y'(t)$ in the following form.

$$y'(t) = \overline{y'} + y_1' e^{-(\theta+c)t} + y_2' e^{-\gamma t} \tag{17}$$

for $t \geq t_0$, where $\overline{y'}$, y_1' and y_2' are some constants. $\overline{y'}$ corresponds to the equilibrium value of $y'(t)$. Since we know that the equilibrium point is in area I, we have $\overline{y'} > 0$. At time $t=t_0$, the system enters area I from area II. So obviously, we should have $\frac{dy'(t_0)}{dt} > 0$.

Hence, the initial condition for Eq. (17) is $y'(t_0) = 0$ and $\frac{dy'(t_0)}{dt} > 0$. It is easy to see that $y'(t)$ has at most one extremum point for $t > t_0$. If a time t_1 exists such that $y'(t_1) < 0$, then $y'(t)$ will have at least two extremum points. So $y'(t) \geq 0$ for all $t > t_0$ and the system is always in area I after t_0.

Similarly, if $\gamma = \theta + c$, we can write $y'(t)$ as

$$y'(t) = \overline{y'} + y_1' e^{-\gamma t} + y_2' t e^{-\gamma t}$$

for $t>t_0$. Again, $y'(t)$ has at most one extremum point for $t>t_0$ and we can prove that $y'(t) \geq 0$ for all $t>t_0$.

.

So, when the downloading bandwidth is the bottleneck, we prove that after a finite time t_0, the system will always be in area I and hence, the system is globally stable.

Similarly, when $\frac{1}{c} = \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})$, we have the following lemma. Note from the definition, area I ($y \leq kx$) includes the line $y=kx$.

Lemma 2 If $\frac{1}{c} = \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})$, the system will stay in area I or area II forever after a finite time and system is globally stable.

Proof: When $\frac{1}{c} = \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})$, the equilibrium point is on the line $y=kx$ and it is the equilibrium point of both Eq. (14) and Eq. (15). Note that in this case, $\gamma>\mu$ and hence both Eq. (14) and Eq. (15) are stable linear systems. So if after a finite time, the system is always in area I or area II and never switches again, the system is globally stable. If not, we assume that at time t_0, the system enters area I from area II. Similarly to the proof of

Lemma 1, we can prove that the system will never go back to area II after time t_0. Note that the only difference is $\overline{y'}=0$ here and it will not affect the proof. Hence, the system is also globally stable in this case.

When $\frac{1}{c} < \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})$, the uploading bandwidth is the bottleneck. The equilibrium point is in area II and the linear system Eq. (15) is stable. If both eigenvalues of the system Eq. (15) are real, we can use the same approach as before to prove the global stability. However, when the eigenvalues of the system Eq. (15) are complex, the analysis becomes much more complicated. We will need to introduce a Lyapunov function to study the system. Without loss of generality, we assume that the system is initially in area II. Note that if the system is initially in area I, we can always wait for it to enter area II and study the system from then on. Let t_{2i}, $i = 0, 1, \cdots$ be the time point that the system enters area I from area II the $(i+1)$th time, t_{2i+1} be the time point that the system enters area II from area I the $(i+1)$th time. Then at t_0, the system enters area I from area II the first time, and at t_1, the system enters area II from area I the first time. If we have a finite set of such time points, then the system always stays in area II after a finite time and the system is obviously stable. So, next we will prove the set of $\{t_{2i}\}$ is finite. We first introduce two useful lemmas.

Lemma 3 If $\frac{1}{c} < \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\Upsilon})$, for any i=0, 1, …, $|\dot{y}'(t_{2i+1})| \le |\dot{y}'(t_{2i})|$, where $\dot{y}'(t) = \frac{dy'(t)}{dt}$.

Proof: Without loss of generality, we will prove $|\dot{y}'(t_1)| \le |\dot{y}'(t_0)|$. Note that in the time period $[t_0, t_1]$, the system is in area I and recall that in area I, $y'(t)$ takes the following form,

$$y'(t) = \overline{y'} + y_1'e^{-(\theta+c)t} + y_2'e^{-\gamma t} \quad \text{if } \gamma \ne \theta + c$$

$$y'(t) = \overline{y'} + y_1'e^{-\gamma t} + y_2'te^{-\gamma t} \quad \text{if } \gamma = \theta + c,$$

where $\overline{y'}$, y_1' and y_2' are all are some constants.

Let $z=[\,y'(t) - \overline{y'}, \dot{y}'(t)\,]^T$, since in area I, the system is linear, we will have

$$\frac{dz(t)}{dt} = Az(t),$$

where the matrix A takes the form

$$A = \begin{pmatrix} 0 & 1 \\ a_1 & a_2 \end{pmatrix},$$

and a_1, a_2 are two constant. Since the linear system in area I is stable, the eigenvalues of A should have negative real parts, from which we can derive that a_1<0 and a_2<0.

Now, let

$$P = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix},$$

where $\lambda_1 > 0, \lambda_2 > 0$ are two constants and define the Lyapunov function

$$V(z) = z^T P z = \lambda_1 (y'(t) - \overline{y'})^2 + \lambda_2 (\dot{y}'(t))^2 \tag{18}$$

Then $\dot{V}(z) = -z^T Q z$, where

$$Q = -(A^T P + PA) = \begin{pmatrix} 0 & -(\lambda_1 + \lambda_2 a_1) \\ -(\lambda_1 + \lambda_2 a_1) & -2\lambda_2 a_2 \end{pmatrix}.$$

Since $a_1<0$, $a_2<0$, if we let $\lambda_1 = \lambda_2 a_1$, it is easy to see that $Q \geq 0$ and $\dot{V}(z) \leq 0$. So, $V(z)$ is a non-increasing function of time t. Since $t_1 > t_0$, we have $V(z(t_1)) \leq V(z(t_0))$. Note that at t_0 and t_1, $y'(t_0) = y'(t_1) = 0$. So, from Eq. (18), we have

$$|\dot{y}'(t_1)| \leq |\dot{y}'(t_0)|.$$

Next, we consider the system's behaviour in area II. Without loss of generality, we consider the time period $t \in [t_1, t_2]$. Note that here we are studying the case that the eigenvalues of the system Eq. (15) are complex numbers. So, $y'(t)$ will take the following form,

$$y'(t) = -\overline{y'} + Ae^{-\alpha(t-t_1)} \cos(\omega(t - t_1) + \phi),$$

where $\overline{y'} > 0, A > 0, \alpha > 0, \omega > 0$, and ϕ are some constants. Note that $y'(t_1) = 0$, so $\cos\phi = \frac{y'}{A} > 0$ and we have $-\pi/2 \leq \phi \leq \pi/2$. The derivative of $y'(t)$ can then is shown to be

$$\dot{y}'(t) = -Aae^{-\alpha(t-t_1)} \sin + Ae^{-\alpha(t-t_1)} \cos(\omega(t - t_1) + \phi + \beta),$$

where $a = \sqrt{\alpha^2 + \omega^2}$ and $\beta = \tan^{-1} \frac{\alpha}{\omega}$.

Lemma 4 *If* $\frac{1}{c} < \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\Upsilon})$, *in the time period* $t \in [t_1, t_2]$, *the system is in area II and we have*

1. *There exists a constant* $\varepsilon > 0$. *If* $|\dot{y}'(t_1)| \le \varepsilon$, *then* $t_2=\infty$, *i.e., the system will stay in area* II *forever after* t_1.

2. *If* t_2 *is finite, then*

$$|\dot{y}'(t_2)| \le \left| e^{-\frac{a\pi}{\omega}} \dot{y}'(t_{2i}) \right|.$$

Proof: Let $\hat{t}$ be the first local maximum point of $y'(t)$. Then from $\dot{y}'(\hat{t}) = 0$, we can get $\hat{t} = t_1 + \frac{2\pi - \beta - \phi}{\omega}$ and

$$y'(\hat{t}) = -\overline{y'} + Ae^{-\alpha(\hat{t}-t_1)}\cos\beta. \tag{19}$$

Note that $\dot{y}'(t_1) = -Aa\sin(\phi + \beta)$. When $\dot{y}'(t_1) < Aa\sin\beta$, we have $-\beta \le \phi \le 0$. At time t_1, we also have $y'(t_1) = 0$, i.e., $A\cos\phi = \overline{y'}$. So, $A = \overline{y'}/\cos\phi \le \overline{y'}/\cos\beta$. From Eq. (19), we see that

$$y'(\hat{t}) \le -\overline{y'}(1 - e^{-\alpha(\hat{t}-t_1)}).$$

since $\hat{t} - t_1 \ge \frac{\pi}{\omega}$, $y'(\hat{t})$ is strictly less than zero. So, if we let $\varepsilon = \overline{y'}a\sin\beta$, when $|\dot{y}'(t_1)| \le \varepsilon \le Aa\sin\beta$, $y'(\hat{t}) < 0$, i.e., the system will never go back to area I again and $t_2 = \infty$.

Now, if t_2 is finite, from the previous proof, we know that $0 \le \phi \le \pi/2$. From $y'(t_1) = y'(t_2) = 0$, we have

$$\overline{y'} = A\cos\phi = Ae^{-\alpha(t_2-t_1)}\cos(\omega(t_2-t_1)+\phi).$$

It is easy to see that $\omega(t_2-t_1) \geq 2(\pi-\phi) \geq \pi$. We also have $\frac{3\pi}{2} \leq \omega(t_2-t_1)+\phi \leq 2\pi$.

Hence

$$\omega(t_2-t_1)+\phi+\beta \geq \frac{3\pi}{2}+\beta.$$

Since $t_2 \leq \hat{t}$, we also have $\omega(t_2-t_1)+\phi+\beta \leq 2\pi$. So,

$$\frac{3\pi}{2}+\beta \leq \omega(t_2-t_1)+\phi+\beta \leq 2\pi.$$

Now, if $0 \leq \phi+\beta \leq \pi/2$, we have

$$\frac{3\pi}{2} \leq 2\pi-(\phi+\beta) \leq 2\pi.$$

and from $\omega(t_2-t_1) \geq 2(\pi-\phi)$, we know

$$\omega(t_2-t_1)+\phi+\beta \geq 2(\pi-\phi)+\phi+\beta = 2\pi+2\beta-(\phi+\beta) \geq 2\pi-(\phi+\beta).$$

Hence

$$|\sin(\omega(t_2-t_1)+\phi+\beta| \leq |\sin(2\pi-(\phi+\beta))| = |\sin(\phi+\beta)|.$$

If $\pi/2 \leq \phi+\beta \leq \pi$, then

$$\frac{3\pi}{2} \leq \pi+(\phi+\beta) \leq 2\pi.$$

Since $\omega(t_2-t_1) \geq \pi$, we have $\omega(t_2-t_1)+\phi+\beta \geq \pi+\phi+\beta$ and

$$|\sin(\omega(t_2-t_1)+\phi+\beta)| \leq |\sin(\pi+\phi+\beta)| = |\sin(\phi+\beta)|.$$

So, in both cases, we always have $|\sin(\omega(t_2-t_1)+\phi+\beta)| \leq |\sin(\phi+\beta)|$. And finally, we have

$$\left|\frac{\dot{y}'(t_2)}{\dot{y}'(t_1)}\right| = \left|\frac{Aae^{-\alpha(t_2-t_1)}\sin(\omega(t_2-t_1)+\phi+\beta)}{Aa\sin(\phi+\beta)}\right|$$

$$\leq e^{-\alpha(t_2-t_1)}$$

$$\leq e^{-\frac{\alpha\pi}{\omega}} < 1.$$

Now, with the help of Lemmas 3 and 4, we will also have the global stability for the third case $\frac{1}{c} < \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})$.

Lemma 5 If $\frac{1}{c} < \frac{1}{\eta}(\frac{1}{\mu} - \frac{1}{\gamma})$, the system will stay in area II forever after a finite time and the system is globally stable.

Proof: From Lemmas 3 and 4, we have

$$\left|\dot{y}'(t_{2i+1})\right| \leq \left|\dot{y}'(t_{2i})\right| \leq e^{-\frac{\alpha\pi}{\omega}}\left|\dot{y}'(t_{2i-1})\right| \leq e^{-\frac{i\alpha\pi}{\omega}}\left|\dot{y}'(t_1)\right|.$$

So, for a given $\varepsilon = \overline{y}' a \sin\beta$, when i is large enough, we have $\left|\dot{y}'(t_{2i+1})\right| \leq \varepsilon$ and from Lemma 4, after t_{2i+1}, the system will stay in area II forever. Hence the system is globally stable.

Finally, combining the three cases from Lemmas 1, 2, and 5, we have our main theorem.

Theorem 1 For any initial condition x(0)>0 and y(0)≥0, the fluid model described by Eq. (12) is always globally stable.

Note that Theorem 1 is a very strong result. It is true for any reasonable network parameters λ, c, μ, η, θ and γ. The only exception that the system is unstable is when γ=0, which means that seeds will never leave the system and it will not happen in any real network. The result also implies that BitTorrent-like P2P systems are inherently stable and agrees with many observations in real BitTorrent networks. Hence it provides a solid theoretical support for steady state performance analysis of P2P networks.

3.5 NUMERICAL RESULTS

We numerically simulated the fluid model with different parameter settings and different initial conditions. In all of our simulations, we find that the system converges eventually, although it may take different time to converge depending on the settings. Note that validation of the fluid model has been done in [37]. Hence here we only study the stability of the fluid model and will not compare the fluid model with the Markov model or real trace of BitTorrent-like networks.

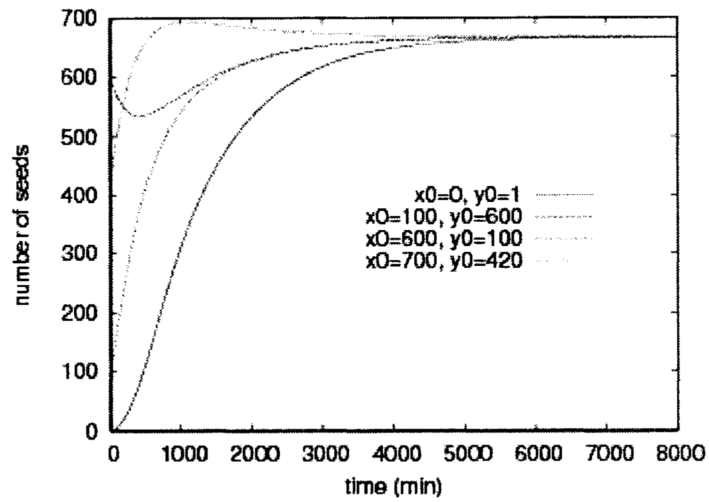

Figure 3.2: The evolution of the number of seeds (downloading bandwidth constraint)

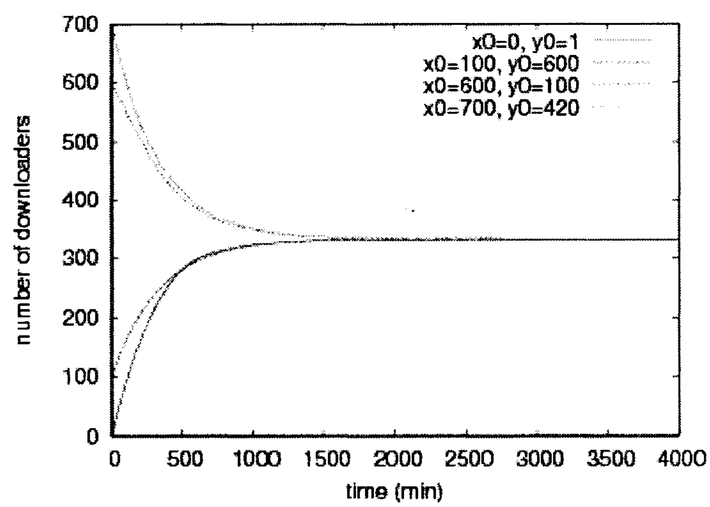

Figure 3.3: The evolution of the number of downloaders (downloading bandwidth constraint)

In Figures 3.2 and 3.3, we simulated the fluid model under different initial conditions. We chose the following parameters for this simulation: $\lambda=1$, μ=0.00125, c=0.002, $\theta=\gamma$=0.001. We also set $\eta=1$. This is in keeping with our observation regarding the efficiency of the download as described in [37]. The time unit is one minute. Note that with this setting, we have k=0.6 and the line y=0.6x is the boundary of area I and area II.

From Eq. (13), we can find the equilibrium point in this case is $\bar{x}$=333 and $\bar{y}$=667. We compared the system under four different initial settings. The initial number of downloaders x_0 and the number of seeds y_0 are shown in the figures. In the first initial setting, we have x_0=0 and y_0=1, which is the normal case when a new P2P system starts with one seed and no downloaders. The second initial setting x_0=20, y_0=600 and the third initial setting x_0=600, y_0=20 fall into area I and area II respectively. The last setting y_0=700, y_0=420 falls exactly on the line $y=kx$, which is the boundary of area I and area II. From the figures, we can see that the number of downloaders becomes steady when time is around 2000 minute, while the number of seeds takes a much longer time, at around 6000 minute, to converge. Overall, the whole system converges to the equilibrium point and remains in the steady state after t=6000. In this simulation, since $\gamma<\mu$, we know that downloading bandwidth is the bottleneck and the equilibrium point is in area I.

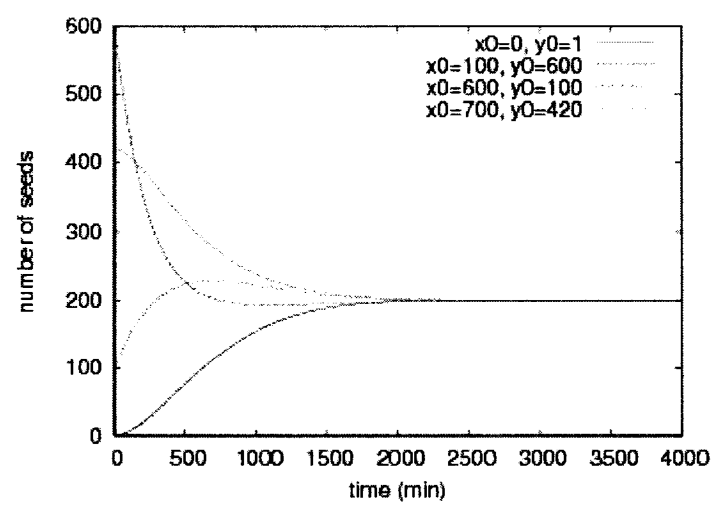

Figure 3.4: The evolution of the number of seeds (uploading bandwidth constraint)

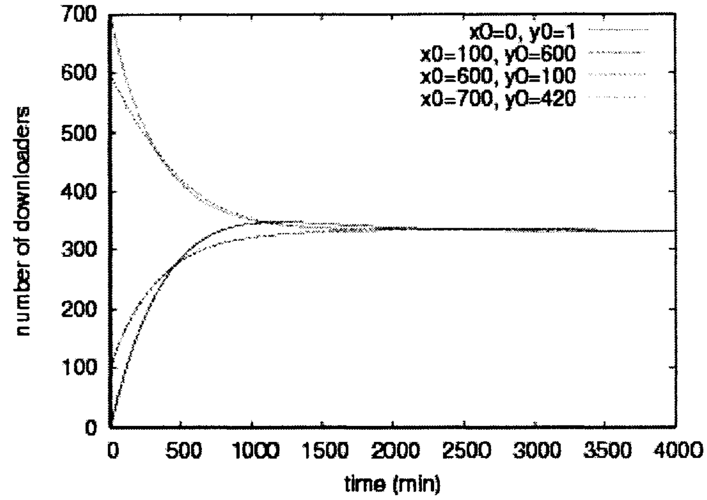

Figure 3.5: The evolution of the number of downloaders (uploading bandwidth constraint)

In Figures 3.4 and 3.5, we have the same setting as the first simulation, except that now we set γ=0.005. With the change of γ, the new equilibrium point is $\bar{x}$=375 and $\bar{y}$=125. The boundary line is the same as before y=0.6x. Now, the uploading bandwidth becomes the bottleneck and the equilibrium point falls into area II. In this setting, we have the similar result as before. After t=2000, the whole system reaches the equilibrium point.

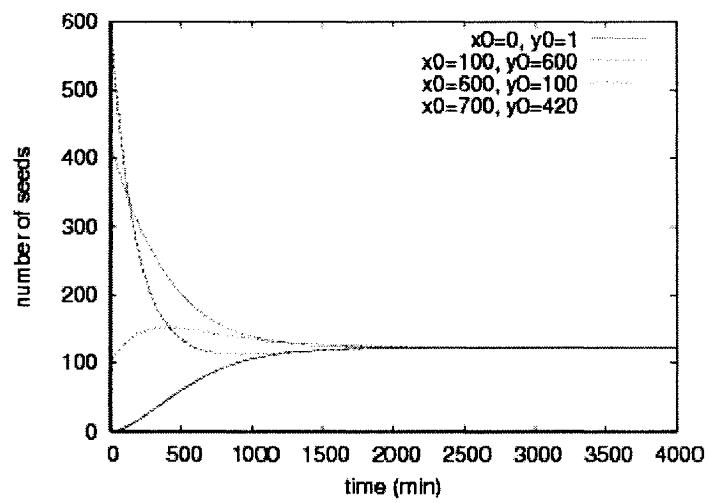

Figure 3.6: The evolution of the number of seeds (switching line)

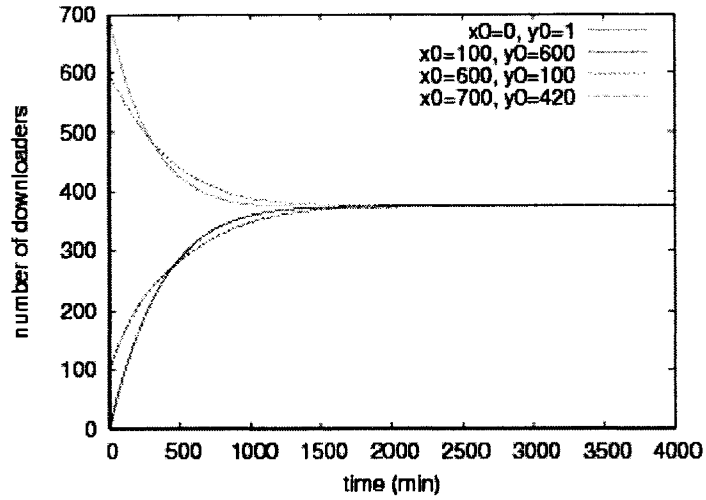

Figure 3.7: The evolution of the number of downloaders (switching line)

In the last simulation (shown in Figures 3.6 and 3.7), we set γ=0.00333.

Hence $\frac{1}{c}=\frac{1}{\eta}(\frac{1}{\mu}-\frac{1}{\gamma})$, and the equilibrium point $\bar{x}$=333, $\bar{y}$=200 is on the line y=0.6x.

Again, we see that the system becomes stable after t=3000.

From these simulations, we see that no matter where the equilibrium point is, in area I, in area II, or on the line $y=kx$, the system converges under any given initial conditions. Hence, it verifies our result that the fluid model is globally stable.

3.6 CONCLUSION

In this chapter, we used the fluid model to study the stability of P2P file sharing systems. The fluid model of BitTorrent-like file sharing networks is a switched linear system. By using Lyapunov function, we prove that such a switched linear system is always globally stable and hence provide a theoretical support for the steady state performance analysis of P2P networks. We also verified our result by extensive simulations.

The model we used for the stability study is relatively simple. For example, all peers are assumed to have the same downloading and uploading bandwidth. In [37], it has been shown that this simple mdel can capture the fundamental behaviours of BitTorrent networks. However, for more accurate performance analysis, we need to take consideration of the peer heterogeneousness. The stability issue in such a more realistic scenario is part of our future work.

CHAPTER 4 CONCLUSIONS AND FUTURE WORK

4.1 CONCLUSIONS

The design of many new Internet applications is based on the peer-to-peer model. Among those applications, P2P file sharing has been one of the most popular applications. In this thesis, the efficiency and stability of BitTorrent-like P2P file sharing systems have been studied.

In chapter 2, motivated by the fact that the upload bandwidth is the most important resource in a P2P file sharing network, a stochastic model has been proposed to study the peer distribution with regard to the number of pieces that a peer has. Under the assumptions that the network is large and is in a steady state, a set of equations are derived to describe the system. By numerically solving these equations, we are able to obtain the peer distribution. Once the peer distribution is known, the performance of the system can be analyzed. For example, by applying Little's Law, we can obtain the average download time of the system. More importantly, through this model, we are able to understand how the performance of the system is affected by different network parameters such as the seed departure rate, the piece number, and the number of neighbours etc. We find that under different conditions, the influence of these parameters is different to the performance. In some cases, a small change in the parameters may affect the performance significantly. While in other cases, changes in parameters may still affect the performance, but not significantly. Based on these results, we get some useful guidelines on how to design an efficient P2P file sharing network.

In chapter 3, a theoretical analysis for the stability of BitTorrent-like P2P file sharing systems has been proposed. There is a lot of research on the steady state performance analysis of P2P file sharing networks. However, theoretically, it is still not very clear to us whether such a steady state always exists. Motivated by this fact, the fluid model proposed in [37] has been used to study the global stability of BitTorrent-like P2P systems. The fluid model proposed in [37] turns out to be a switched linear system, which in general, is very hard to determine the global stability. Fortunately, for the fluid model, we are able to prove that it is always globally stable under any reasonable initial conditions. Our theoretical results have also been verified by extensive simulations. In all the simulations, the system converges to the equilibrium point and hence a strong theoretical support is provided for the steady state performance analysis of P2P networks.

4.2 FUTURE WORK

Although the models used in this thesis can capture all the important features of P2P file sharing systems, they are still relatively simple because of the analysis simplification. For example, in both models, we assume that all peers have the same upload bandwidth. However, in real networks, peers may have different upload bandwidth and to make things more complicated, the available bandwidth may even change with time. One of the future works is to make the model more realistic, for example, to take the peer heterogeneousness into account in the model. In addition, although we obtained some useful guidelines on how to design an efficient P2P file sharing network in chapter 2, it is still not very clear to us how to choose the network parameters to optimize the performance in general cases. For example, we have shown that when the number of neighbours is around six, the system perform the best in our simulations. However, if other parameters like piece number, seed departure rate etc are changed, how should we change the number of neighbours so that we can still maintain the best performance? To answer these questions is also possible future work. Finally, in this thesis, the numerical results are all obtained through simulations. We haven't been able to do experiments in real BitTorrent networks to test our results due to the complexity of the experiments. To extensively verify our results in real networks is also part of our future work.

REFERENCES

[1] Robert Kaye, "Next-Generation File Sharing with Social Networks", *O'Reilly Emerging Technology Conference*, May 2004.

[2] Vagan Terziyan, Andriy Zharko, "Semantic Web and Peer-to-Peer: Integration and Interoperability", *International Journal of Computers, Systems and Signals*, Vol. 4, No. 2, pp. 33-46, 2003.

[3] David Barkai, "An Introduction to Peer-to-Peer Computing", *Intel Developer UPDATE Magazine*, pp. 1-7, February 2000.

[4] Donna Wolff, "Networking Definitions - Peer-to-peer", *SearchNetworking.com*, http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci212769,00.html, Febrary 2004.

[5] Wikipedia, "Node (networking)", http://en.wikipedia.org/wiki/Node_%28networking%29, (Current August 20, 2007).

[6] Ellacoya Networks, "Ellacoya Data Shows Web Traffic Overtakes Peer-to-Peer (P2P)", http://www.ellacoya.com/news/pdf/2007/NXTcommEllacoyaMediaAlert.pdf, June 18, 2007.

[7] Nate Anderson, "The YouTube effect: HTTP traffic now eclipses P2P", *Ars Technica*, http://arstechnica.com/news.ars/post/20070619-the-youtube-effect-http-traffic-now-eclipses-p2p.html, June 19, 2007.

[8] Philip Miller, "TCP/IP Explained", *Digital Press*, pp. 313-342, March 1997.

[9] Carles Pairot Gavaldà, Pedro García López, and Rubén Mondéjar Andreu, "Deploying Wide-Area Applications Is a Snap", *IEEE Internet Computing,* Vol. 11, No. 2, pp. 72-79, March 2007.

[10] Can Erten, Richard MacManus, "P2P: Introduction and real World application", *read/writeweb,* http://www.readwriteweb.com/archives/p2p_introduction_real_world_applications.php, March 29, 2007.

[11] Wikipedia, "Peer-to-peer", http://en.wikipedia.org/wiki/Peer-to-peer, (Current August 20, 2007).

[12] Wikipedia, "File sharing", http://en.wikipedia.org/wiki/File_sharing, (Current August 20, 2007).

[13] Wikipedia, "Kazaa", http://en.wikipedia.org/wiki/Kazaa, (Current August 20, 2007).

[14] Wikipedia, "KCeasy", http://en.wikipedia.org/wiki/KCeasy, (Current August 20, 2007).

[15] "Napster messages", http://opennap.sourceforge.net/napster.txt, April 7, 2000.

[16] Green, Matthew, "Napster Opens Pandora's Box: Examining How File-Sharing Services Threaten the Enforcement of Copyright on the Internet", *Ohio State Law Journal,* Vol. 63, No. 2, 2002.

[17] Wikipedia, "EDonkey network", http://en.wikipedia.org/wiki/EDonkey_network, (Current August 20, 2007).

[18] Wikipedia, "Overnet", http://en.wikipedia.org/wiki/Overnet, (Current August 20, 2007).

[19] Wikipedia, "FastTrack", http://en.wikipedia.org/wiki/FastTrack, (Current August 20, 2007).

[20] "The FastTrack Protocol", http://cvs.berlios.de/cgi-bin/viewcvs.cgi/gift-fasttrack/giFT-FastTrack/PROTOCOL?rev=HEAD&content-type=text/vnd.viewcvs-markup, (Current August 20, 2007).

[21] Ghosemajumder, Shuman, *Advanced Peer-Based Technology Business Models*, Massachusetts Institute of Technology, Sloan School of Management, 2002.

[22] Wikipedia, "LimeWire", http://en.wikipedia.org/wiki/LimeWire, (Current August 20, 2007).

[23] Wikipedia, "Imesh", http://en.wikipedia.org/wiki/IMesh, (Current August 20, 2007).

[24] Wikipedia, "BitComet", http://en.wikipedia.org/wiki/BitComet, (Current August 20, 2007).

[25] Wikipedia, "Emule", http://en.wikipedia.org/wiki/EMule, (Current August 20, 2007).

[26] ipoque GmbH, "P2P Survey 2006 – Extended Abstract", http://www.ipoque.com/pub/P2P-Survey-2006.pdf, http://www.ipoque.com/en/pressrelease_ipoque_241006.html, (Current August 20, 2007).

[27] T. S. E. Ng, Y.-H. Chu, S. G. Rao, K. Sripanidkulchai, and H. Zhang, "Measurement-based Optimization Techniques for Bandwidth-Demanding Peer-To-Peer Systems", *IEEE INFOCOM*, Vol. 3, pp. 2199-2209, April 2003.

[28] M. Ripeanu, I. Foster, and A. Iamnitchi, "Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design", *IEEE Internet Computing Journal*, Vol. 6, No. 1, 2002.

[29] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley, "Modeling peer-peer file sharing systems", *IEEE INFOCOM*, Vol. 3, pp. 2188-2198, 2003.

[30] F. Clevenot and P. Nain, "A Simple Fluid Model for the Analysis of the Squirrel Peer-to-Peer Caching System", *IEEE INFOCOM*, Vol. 1, pp. 86-95, March 2004.

[31] F. Clevenot, P. Nain, and K. Ross, "Stochastic Fluid Models for Cache Clusters", *Performance Evaluation*, Vol. 59, No. 1, pp. 1-18, January, 2005.

[32] D. G. Luenberger, "Introduction to Dynamic Systems: Theory, Models, and Applications", *John Wiley & Sons*, 1979.

[33] Little, J. D. C, "A Proof of the Queueing Formula $L = \lambda W$", *Operations Research*, Vol. 9, No. 3, pp. 383-387, 1961.

[34] Bram Cohen, "Incentives Build Robustness in BitTorrent", http://bitconjurer.org/BitTorrent/bittorrentecon.pdf, May 2003.

[35] Dongyu Qiu, Wei Qian Sang, "Global Stability of Peer-to-Peer File Sharing Systems", accepted by *Elsevier Computer Communications*, 2007.

[36] G. de Veciana and X. Yang. Fairness, "Incentives and Performance In Peer-to-peer Networks", in *the Forty-first Annual Allerton Conference on Communication, Control and Computing, Monticello, IL*, October 2003.

[37] D. Qiu and R. Srikant, "Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks", in *Proceedings of ACM SIGCOMM*, August 2004.

[38] X. Yang and G. de Veciana, "Service Capacity of Peer to Peer Networks", in *Proceedings of IEEE INFOCOM*, 2004.

[39] Wikipedia, "Little's law", http://en.wikipedia.org/wiki/Little's_law, (Current August 20, 2007).

[40] Weiqian Sang, Dongyu Qiu, "On the Efficiency of Peer-to-Peer File Sharing", in *Proceedings of IEEE ICME,* pp. 32-35, July 2007.

[41] M. Ripeanu, "Peer-to-peer architecture case study: Gnutella network," in *Proceedings of International Conference on Peer-to-peer Computing*, August 2001.

[42] Y. Tian, D. Wu, and K. W. Ng, "Modeling, Analysis and Improvement for BitTorrent-Like File Sharing Networks," in *Proceedings of IEEE INFOCOM*, pp. 1-11, April 2006.

[43] S. P. Meyn and R. L. Tweedie, "Markov Chains and Stochastic Stability", *Springer-Verlag*, 1993.

[44] A. Binzenhofer, D. Staehle and R. Henjes, "On the Stability of Chord-based P2P Systems", in *Proceedings of IEEE GLOBECOM*, December 2005.

[45] D. Liberzon and A. Morse, "Basic Problems in Stability and Design of Switched Systems", in *Proceedings of IEEE Control Systems Magazine*, Vol. 19, No. 5, pp. 59–70, October 1999.

[46] Riikka Susitaival, Samuli Aalto, and Jorma T. Virtamo, "Analyzing the Dynamics and Resource Ssage of P2P File Sharing by a Spatio-temporal Model", in *International Conference on Computational Science*, pp. 420–427, May 2006.

[47] Riikka Susitaival and Samuli Aalto, "Modelling the Population Dynamics and the File Availability in a Bittorrent-like P2P System With Decreasing Peer Arrival Rate", in *International Workshop on Self-Organizing Systems (IWSOS)*, pp. 34–48, September 2006.

[48] Lei Guo, Songqing Chen, Zhen Xiao, Enhua Tan, Xiaoning Ding, and Xiaodong Zhang, "Measurements, Analysis, and Modeling of Bittorrent-like Systems", in *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement (IMC)*, pp. 35–48, October 2005.

APPENDIX A: Lyapunov Function

1. Definition of a Lyapunov candidate function

Let

$$V : \mathbb{R}^n \to \mathbb{R}$$

be a scalar function.

V is a Lyapunov-candidate-function if it is a locally positive-definite function, i.e.

$$V(0) = 0$$

$$V(x) > 0 \quad \forall x \in U$$

With U being a neighborhood region around x = 0

2. Definition of the equilibrium point of a system

Let

$$g : \mathbb{R}^n \to \mathbb{R}^n$$

$$\dot{y} = g(y)$$

be a arbitrary autonomous dynamical system with equilibrium point y^*:

$$0 = g(y^*)$$

There always exists a coordinate transformation $x = y - y^*$, such that:

$$\dot{x} = g(x + y^*) = f(x)$$

$$0 = f(x^*) \quad \Rightarrow \quad x^* = 0$$

So the new system f(x) has an equilibrium point at the origin.

3. Basic Lyapunov theorems for autonomous systems

Let

$$x^* = 0$$

be an equilibrium of the autonomous system

$$\dot{x} = f(x)$$

And let

$$\dot{V}(x) = \frac{\partial V}{\partial x}\frac{dx}{dt} = \nabla V \dot{x} = \nabla V f(x)$$

be the time derivative of the Lyapunov-candidate-function V.

Stable equilibrium

If the Lyapunov-candidate-function V is locally positive definite and the time derivative of the Lyapunov-candidate-function is locally negative semidefinite:

$$\dot{V}(x) \leq 0 \quad \forall x \in \mathcal{B}$$

for some neighborhood $\mathcal{B}$, then the equilibrium is proven to be stable.

Locally asymptotically stable equilibrium

If the Lyapunov-candidate-function V is locally positive definite and the time derivative of the Lyapunov-candidate-function is locally negative definite:

$$\dot{V}(x) < 0 \quad \forall x \in \mathcal{B}$$

for some neighborhood $\mathcal{B}$, then the equilibrium is proven to be locally asymptotically stable.

Globally asymptotically stable equilibrium

If the Lyapunov-candidate-function V is globally positive definite and the time derivative of the Lyapunov-candidate-function is globally negative definite:

$$\dot{V}(x) < 0 \quad \forall x \in \mathbb{R}^n,$$

then the equilibrium is proven to be globally asymptotically stable.

From Wikipedia, the free encyclopedia,

http://en.wikipedia.org/wiki/Lyapunov_function.

APPENDIX B: Little's Law

1. Definition

In queuing theory, Little's result, theorem, lemma, or law says:

The average number of customers in a stable system (over some time interval), N, is equal to their average arrival rate, λ, multiplied by their average time in the system, T, or:

$$N = \lambda T.$$

Although it looks intuitively reasonable, it's a quite remarkable result, as it implies that this behavior is entirely independent of any of the detailed probability distributions involved, and hence requires no assumptions about the schedule according to which customers arrive or are serviced. The only assumption is that the system operates in a first-come-first-served manner (FCFS).

It is also a comparatively recent result - it was first proved by John Little, an Institute Professor and the Chair of Management Science at the MIT Sloan School of Management, in 1961.

Handily his result applies to any system, and particularly, it applies to systems within systems. So in a bank, the queue might be one subsystem, and each of the tellers another subsystem, and Little's result could be applied to each one, as well as the whole thing.

The only requirement is that the system is stable -- it can't be in some transition state such as just starting up or just shutting down.

2. Mathematical formalization of Little's theorem

Let $\alpha(t)$ be the arrival rate to some system in the interval [0, t]. Let $\beta(t)$ be the number of departures from the same system in the interval [0, t]. Both $\alpha(t)$ and $\beta(t)$ are integer valued increasing functions by their definition. Let T_t be the mean time spent in the system (during the interval [0, t]) for all the customers who were in the system during the interval [0, t]. Let N_t be the mean number of customers in the system over the duration of the interval [0, t].

If the following limits exist,

$$\lambda = \lim_{t\to\infty} \alpha(t)/t,$$

$$\delta = \lim_{t\to\infty} \beta(t)/t,$$

$$T = \lim_{t\to\infty} T_t,$$

and, further, if $\lambda = \delta$ then Little's theorem holds, the limit

$$N = \lim_{t\to\infty} N_t,$$

exists and is given by Little's theorem,

$$N = \lambda T.$$

From Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Little's_law.