HYBRID CDN P2P ARCHITECTURE FOR MULTIMEDIA STREAMING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

KASIM ÖZTOPRAK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

FEBRUARY 2008

Approval of the thesis:

HYBRID CDN P2P ARCHITECTURE FOR MULTIMEDIA STREAMING

submitted by **KASIM ÖZTOPRAK** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan ÖZGEN
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Volkan ATALAY
Head of Department, **Computer Engineering** ________________

Prof. Dr. Volkan ATALAY
Supervisor, **Computer Engineering Dept., METU** ________________

Assoc. Prof. Dr. Gözde Bozdağı AKAR
Co-Supervisor, **Electrical and Electronics Eng. Dept., METU** ________________

Examining Committee Members:

Prof. Dr. Müslim BOZYİĞİT
Computer Engineering Dept., METU ________________

Prof. Dr. Volkan ATALAY
Computer Engineering Dept., METU ________________

Assoc. Prof. Dr. Ahmet COŞAR
Computer Engineering Dept., METU ________________

Assoc. Prof. Dr. Y. Murat ERTEN
Computer Engineering Dept., TOBB-ETU ________________

Asst. Prof. Dr. Cüneyt BAZLAMAÇCI
Electrical and Electronics Engineering Dept., METU ________________

Date: 08/02/2008

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Lastname :

Signature :

ABSTRACT

HYBRID CDN P2P ARCHITECTURE FOR MULTIMEDIA STREAMING

ÖZTOPRAK, Kasım

Ph.D., Department of Computer Engineering

Supervisor: Prof. Dr. Volkan ATALAY

Co-Supervisor: Assoc. Prof. Dr. Gözde Bozdağı AKAR

February 2008, 116 pages

In this thesis, the problems caused by peer behavior in peer-to-peer (P2P) video streaming is investigated. First, peer behaviors are modeled using two dimensional continuous time markov chains to investigate the reliability of P2P video streaming systems. Then a metric is proposed to evaluate the dynamic behavior and evolution of P2P overlay network. Next, a hybrid geographical location-time and interest based clustering algorithm is proposed to improve the success ratio and reduce the delivery time of required content. Finally, Hybrid Fault Tolerant Video Streaming System (HFTS) over P2P networks has been designed and offered conforming the required Quality of Service (QoS) and Fault Tolerance. The results indicate that the required QoS can be achieved in streaming video applications using the proposed hybrid approach.

Keywords: peer-to-peer video streaming, clustering, fault tolerance, channel capacity, peer-to-peer metrics

ÖZ

ÇOĞULORTAM İLETİŞİM İÇİN MELEZ CDN P2P YAPISI

ÖZTOPRAK, Kasım

Doktora, Bilgisayar Mühendisliği Bölümü

Tez Danışmanı: Prof. Dr. Volkan ATALAY

Ortak Tez Danışmanı: Doç Dr. Gözde Bozdağı AKAR

Şubat 2008, 116 sayfa

Bu tezde eşler arası video akışı uygulamalarında eş davranışından kaynaklanan sorunlar incelenmiştir. Ilk önce, video akışı uygulamalarının güvenilirliğini incelemek amacı ile eş davranışları iki boyutlu sürekli markov zinciri kullanılarak modellenmiştir. Daha sonra eşler arası oluşan ağın değişen davranışlarını ve gelişimini ölçebilmek için bir ölçü birimi önerilmiştir. Sonra, sistemde yapılan aramaların başarım oranını artırmak ve istenen verilerin dağıtım süresini azaltmak amacı ile coğrafi lokasyon-zaman ve ilgi temelli melez bir kümeleme algoritması önerilmiştir. Son olarak, istenilen servis kalitesini ve sürekliliği karşılayabilen, eşler arası ağ üzerinde çalışan Melez Hata Bağımsız Video Akışı sistemi tasarlanmış ve önerilmiştir. Yapılan çalışmaların sonuçları video akışı uygulamalarında istenilen servis kalitesinin önerilen melez yaklaşım ile sağlanabildiğini götermektedir.

Anahtar Kelimeler: kişiden kişiye video iletimi, kümeleme, süreklilik, kanal kapasitesi, kişiden kişiye iletisim ölçü birimi

ACKNOWLEDGEMENTS

Before all, my thanks go to my supervisors, Prof. Dr. Volkan ATALAY and Assoc. Prof. Dr. Gözde Bozdağı AKAR for everything. I never left their room with disappointment or discouragement. Each of our meetings filled me with determination and will to study.

I wish to express my thanks to my family, for their intense support throughout this study. I am grateful to my colleagues at Atılım University, Computer Engineering Department for providing a researcher-friendly medium.

I thank to Prof. Dr. Müslim BOZYİĞİT and Assoc. Prof. Dr. Y. Murat ERTEN for their kind advises on my studies, to Akif YAZICI for his help on analytical analysis and introducing LaTeXto me, to Assoc. Prof. Dr. Nail AKAR for his insight help on analytical analysis, to Asst. Prof. Dr. Hürevren KILIÇ in contribution for the study of new metric development for P2P architectures, to Çiğdem TURHAN for her kind support in reading the drafts of thesis, and İlteriş ŞULE and Sun Microsystems for their kind support for powerful computing infrastructure used in the simulations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

This chapter introduces the research problems addressed in this thesis, and summarizes our main contributions.

1.1 Introduction

As the telecommunication technology evolved over time, applications requiring higher bandwidth such as audio/video streaming became more and more popular. The popularity and high interest in such applications bring several problems with them. One of the most critical problems of such applications to be solved is the number of potential users to be served. In the early days of audio/video delivery over Internet, the content providers directly deployed the distribution systems in their environments and invested in applications, servers and connections to comply with the expectations of the potential customers around the world. The high interest of the customers around the world causes performance, scalability, and high cost problems for such systems. The solution to the scalability and expensive price problems is using the services provided by the professional third-party companies to reduce the prices and improve the performance of delivery.

The systems established to deliver digital media conforming to the requirements of high bandwidth and faster delivery is called Content Distribution Network (CDN). The key point of such systems is introducing local traffic exchange for: (1) faster delivery of digital medium by using high speed local links, (2) less traffic overhead by keeping the content very close to the clients, (3) supporting higher quality by bringing less variable delay, and low loss rate. In the CDN systems the original copy of the media is kept either in the center of the content owner or the central server farm of the CDN system. A CDN system could have number of servers ranging from a few to thousands of servers distributed around the world. The CDN systems employ several applications to monitor the traffic across the internet and to direct the clients to the appropriate servers especially using the information where they are connecting from.

Direct deployment of video services would be useful for the number of concurrent connections filling up to 155 Mbps or 622 Mbps speeds since an organization or a company can set up an Internet connection with such speeds. The management efforts and costs of such system might cause significant problem to the content owner. On the other hand, when the number of connections increase it would be very costly to implement direct connection in house by the content owner. CDN approach, however, might be acceptable to handle higher bandwidth requirements. Although, the CDN approach is perfect for service quality and the scalability issues, the cost factor prevents the content owners from delivering their audio/video streams through CDN system since they charge for the amount of traffic transmitted. Additionally, as the number of users obtaining services from the CDN system increases, CDN systems cannot answer the requests properly even for poor quality multimedia streams [34].

The evolution in the telecommunication sector during the last decade enabled the users to connect to the Internet through inexpensive broadband communication medium. This results in explosion in the use of peer-to-peer applications. Peer-to-peer systems provide the most scalable computing infrastructure by aggregating the resources from already available peers. It also improves the fault tolerance by removing the dependency to central servers and distributing the load to end systems.

Even though it is gaining high interest in the research and commercial world there are still problems to be solved in P2P systems. These problems could be listed as: (1) The existing P2P frameworks concentrate on the scalability issues and the reduction of the packet delivery or lookup time by enabling location or geographical time based clustering architectures. The reliability or the durability of this service has not been investigated in detail yet, and some form of duplication is assumed to provide the durability of the service. (2) The studies on the reliability of the P2P systems do not investigate the reliability of the P2P systems analytically. Although it provides several ideas on reliability, they do not investigate the requirements of the P2P system. (3) The Quality of the Service (QoS) is only considered to have the packets delivered as early as possible. (4) The current clustering techniques are only evaluated by the search lookup time and there is no detailed research on the capacity of the architecture. The clustering approaches are only concentrating on a single scope, e.g., either interest or geographical location-time. (5) There is no universal metric to evaluate the dynamic evolution and information flow capability of the P2P system.

1.2 Thesis contribution

In this thesis, we developed solutions for the research problems mentioned above. The key ideas of the framework that we offer can be summarized as follows: (1) The reliability of the system is analytically investigated by using a Markov chain–based reliability model for a P2P network having N redundant copies of a requested content using exponential node failure and recovery processes. The model is also enhanced to include backup connections to improve the performance and reliability of the peer-to-peer video streaming systems. (2) A novel hybrid topology and interest based clustering to organize the overlay network in order to reduce startup latency and service interruption probability has been offered. (3) The scalability power of P2P networks have been adapted into streaming systems in order to provide scalable and fault tolerant P2P-CDN video streaming systems. The proposed system increases the availability of the streams with the best quality while keeping the load of the CDN servers under certain limits. The offered model guarantees the fault tolerance and the required Quality of the Services (QoS). (4) To measure the capacity of the P2P architectures more effectively a metric based on Shannon's L-channel capacity calculation idea has been offered. Then, this metric has been compared with the current metrics such as turnover and size change trying to evaluate the performance of the networks. (5) The offered framework for fault tolerant high quality video streaming system is extended to be used in an e-learning system as an application area.

The following sections briefly describe the contributions of the thesis.

1.2.1 Reliability Modeling

Adapting the video streaming applications into P2P framework raised the need for reliability of the streams. In the literature, a method for analytically modeling the P2P streaming systems using fluid methods is proposed in [48]. However, we model the P2P video streaming system in discrete domain.

In [88], a P2P system with increased resilience to peer dynamicity is analyzed. First, a P2P network having N redundant copies of a requested content using exponential node failure and recovery processes is modeled to simulate the nature of a P2P video streaming system. Then by using the designed model, the mean time-to-failure (mttf) for given P2P network parameters such as total number of redundant copies, required number of redundant copies for successful transmission, failure and recovery rates for a peer is derived. Next, the probability of a transmission aborting within a given period of time due to insufficient number of online peers having the content is formulated, and the total number of the peers required in order to sustain an initiated transmission is analyzed. Finally, the Markov model is modified in order to account for the failures of the peers that the client is connected to and the back-up connections.

The analytical study gives an idea about the required number of peers with certain characteristics to guarantee the fault tolerance in video streaming for a specified duration.

1.2.2 Measuring the Capacity of an Overlay Network

The current metrics to measure the performance of the P2P networks are based on the query hit time and query success ratio. Although, these metrics are related with the physical channel characteristics between the peers, the organization of the overlay network, and the behavior of the peers, it does not give an idea on the maximum capacity of the overlay network that can be used. The first major concern of the thesis is to put a new metric to measure the maximum channel capacity for an overlay network to quantitatively compare different overlay networks.

To measure the capacity of the P2P architectures more effectively a metric for P2P networks based on Shannon's L-channel capacity calculation idea has been offered [64,66]. The metric calculates the maximum rate of information (in bits per second) that can be transmitted over a P2P network (a.k.a. combinatorial capacity) caused by protocol and overlay-level connectivity. The suggested method offers to model the P2P systems as a discrete noiseless channel on which the protocol, together with dynamically changing overlay-level instant connectivity topology, defines a Shannon Language. The new metric is used to measure the capacity of the different clustering architectures quantitatively.

To evaluate the reliability of the offered metric, it is first applied to the Gnutella 0.6 protocol for which message traffic explosion and search efficiency is a known problem. Many methods to improve the search efficiency have been proposed. Then the new metric is applied to geographical-time, interest based, and two-way hybrid clustered version of the Gnutella network. The obtained results are compared with the results from other two known metrics' namely, number of query hits and unit query-hit response time and potential correlations among them are examined. The similar metrics measuring the dynamic behavior of the overlay networks during the execution called *turnover* and *size-change* has also been calculated for overlay networks and the results are compared with the offered metric.

1.2.3 Two-Way/Hybrid Clustering Architecture

In the P2P world there are two major clustering techniques currently named as geographical location-time based and interest based clustering. The geographical location-time based clustering model typically clusters the overlay network with respect to the reachability of the peers. In the second approach, the peers are grouped with respect to their interests. All the clusters are formed by using the interest information provided by all peers. A novel interest aware clustering to reduce startup latency and service interruption probability has been offered in this study. The proposed clustering algorithm will also help to organize the overlay network in such a way that it will also benefit from the topology [64].

The new method –Two way/Hybrid Clustering– uses a dispersion method to organize the peers according to both their interests and locality information. The interests of the peers have been considered at the beginning to form the interest clusters while starting to investigate the behaviors of the peers to adaptively organize them into more dynamic overlay networks. In parallel, the peers are clustered in geographical location and access time based manner based on where they are connecting from. This index is also dynamically updated to enlarge the scope of the local network into somewhat bigger local networks by evaluating the RTT values of the neighbor peers in close networks.

In order to evaluate the performance of different clustering architectures, in addition to the average hit time and hit ratio for the searched content, the offered metric has been used.

1.2.4 Hybrid Fault Tolerant Video Streaming

Although, the analytical study shows that with certain number of peers with certain characteristics to guarantee the fault tolerance in video streaming for a specified duration, in reality it does not work properly. Since the models assume threshold for success to a certain level such as %99, the success rate shall be extended to %100 by employing some intelligent algorithms. In order to guarantee the required Quality of Services, we propose hybrid video streaming scheme, which integrates peer to peer video delivery systems into dedicated CDNs together with fault tolerant schemes. The offered method solves the scalability issues of the CDN systems and the unreliability of P2P systems. The key idea in [65] is that the reality still does not match with the analytical studies offered by [48, 50, 88] based on required QoS.

Although, the reliability models promise the fault tolerance, the simulation studies show that the required quality of service (QoS) cannot be obtained from pure P2P systems. Thus, the proposed system uses the CDN servers to introduce fault tolerance by using them as a backup system to locate the required segments of the files while searching for other peers to switch from CDN server in order to keep the load of the CDN servers below specified threshold levels. This way, the CDN servers are used to provide the initial seeds and as the high available failover node to reconstruct the lost connections with the fastest mechanism.

The challenging design of the method uses the contribution of the CDNs not only for initial seeding stage to feed the system with the initial streams but also during the entire operation time. However, we only get a few segments of a failed node which is not available prior to the playout phase in order to guarantee the QoS. The cost caused by adapting the method is very limited since the CDN servers are directly replaced by a new servant (a peer using the resources from the network and sharing the information with other peers within the network) after completing the search.

The method also benefits from the Two-way/hybrid clustering technique by bringing efficient data retrieval with less overhead traffic and fast resource locating and efficient resource usage. The clustering brings fast location and relocation of the content and minimizes the traffic exchange throughout the networks.

1.2.5 Example application area for Multilevel Clustering and HFTS

In this part of the study, the offered two-way/hybrid clustering method is improved to include dual interest cluster index, in addition to the time based cluster index. The learning models are used as the second dimension in the interest level cluster in Three-way/hybrid clustering architecture. By employing the Three-way/hybrid clustering in learning models, the online distance education systems can be optimized for better performance with HFTS algorithm.

Then, we concentrate on extending the modeling of P2P systems with different clustering approaches. The reliability requirements of different clustering architectures, and the effect of organization on the reliability requirements of a P2P system are investigated. It also gives a mechanism to evaluate the different architectures analytically.

1.3 Thesis Organization

Chapter 2 provides a brief introduction to P2P systems. The basic characteristics of the P2P computing paradigm and the available P2P structures are discussed. The difference between the structured and unstructured P2P paradigm is evaluated. The current metrics to measure the performance of P2P systems and the current clustering approaches in P2P are also presented. It then also provides information on rich media distribution and video streaming. The basics of the video streaming over Internet are explained. Finally, the requirements to enable P2P video streaming are also presented.

Chapter 3 presents the modeling of the P2P systems analytically. It starts from scratch to derive the formulae to simulate the nature of a P2P video streaming system to model them. The mean time-to-failure (mttf) for given P2P network parameters such as total number of redundant copies, required number of redundant copies for successful transmission, failure and recovery rates for a peer is derived.

Chapter 4 presents the new metric to measure the channel capacity for overlay networks. The chapter starts out with the presentation of the quantitative metrics to measure the performance of P2P systems. It then introduces the new metric based on Shannon's L-channel capacity calculation idea to monitor the capacity change in an overlay network dynamically throughout the simulations.

Chapter 5 begins with a review of the current clustering approaches: (1) the interest and (2) geographical location-time based clustering algorithms in the literature. The design of new clustering technique is presented which uses the dispersion algorithm to disseminate the data into system in order to locate the best suitable supplier for a requested data when looked up. Then the Hybrid Fault Tolerant Video Streaming System (HFTS) is introduced to provide a reliable video streaming with required QoS.

Chapter 6 extends the studies to application area. Then, the newly designed Fault Tolerant, highly scalable platform for P2P video streaming is applied into educational model by inventing a Three-way/hybrid clustering architecture to be used in e-Learning platforms. Finally, the reliability model offered in chapter 4 is applied to all the known clustering algorithms to analyze their performances quantitatively.

Finally, Chapter 7 concludes the thesis and discusses possible directions for extending this research.

CHAPTER 2

BACKGROUND

This chapter provides basic literature review on peer-to-peer (P2P) systems. The basic definitions, main features and the characteristics of P2P systems are described. Then, the classification of P2P systems and clustering the substrates are explained. After giving information on P2P system structures, peer-to-peer video streaming is introduced. Peer motivation for contribution, failover scenarios, redundancy and suitable video coding for P2P systems are explained briefly.

2.1 Introduction

The evolution in the telecommunication area during the last decade enabled the users to connect to the Internet through inexpensive broadband communication medium. This results in explosion in the use of peer-to-peer applications. Peer-to-peer systems provide the most scalable computing infrastructure by aggregating the resources from already available peers. It also improves the fault tolerance by removing the dependency to central servers and distributing the load to end systems. According to [19], Napster [8] is the first P2P application attracted millions of users around the world. The strength of Napster comes from providing a medium to utilize the idle resources of the connected peers around the world by providing a file sharing mechanism worldwide. In order to locate files in the shared space, a central directory keeping the index of all files are used. The storage was decentralized and functioning P2P style, while locating a file was centralized.

The P2P systems provides glue to stick together all the peers to form an enormous amount of storage, communication medium, and cpu-cycle power.

The role of P2P system is to coordinate and manage the peers for cooperation to form a distributed storage medium, a huge amount of file library for different interests including applications and audio/video, tremendous computing power by sharing their cpu-cycles, and communication links to provide video on demand, real time broadcast and video conferencing systems. The P2P systems would benefit from a centralized server (or group of servers) to provide a coordination mechanism and keep an index of all the contents as in the case of napster or there is no centralized authority to coordinate, control and maintain the entire system including the peers as in gnutella system.

The wide use of P2P applications consumes almost all the communication resources provided by the telecommunication operators for Internet exchange.

2.2 Characteristics of P2P Systems

P2P systems are special kinds of distributed systems where the peers are located in different locations using the Internet as the communication medium among them. The coordination of the components of a P2P system is provided by message passing. The P2P systems can be used to provide networked storage area for keeping the files, putting enormous number of computers to benefit from their cpu cycles and to bring redundancy by having the systems in different geographical locations for the aim of Fault Tolerance. In any P2P system there exists an appropriate protocol to coordinate, control and maintain the entire system depending on the nature of the system.

The characteristics of a P2P system can be summarized in three main categories:

- A peer can join the system from anywhere, anytime regardless of the geographical location and time. There is no governing body to control and manage the join and leave time of a peer. Since the number of peers and the location of them are dynamically changing over time in an overlay network, the topology of the network and connections among peers are highly dynamic.
- A peer can decide to share its resources or not. The resources it has and

contributes may not be known in advance. Additionally, the measurement study carried in [71] shows that peers do not have enough capabilities to act as a server without aggregation because of their high failure ratio and limited capacity. The aggregation of the servers increases total capacity of the resources to be served and increases the reliability of the served content by improving the existence probability of the service.

- The peers are joining to the system to benefit from the resources of the other peers. They can join to, or leave from the system any time, initiate, answer or forward a query according to its own objectives. As noted by [79] the objective of a peer may conflict with the performance of the entire P2P system.

Additionally, there are several other issues that might be considered in a P2P system including: handling heterogeneity to accept contribution with any rate, openness to enable the evolution of the system for new applications, security in order to have the system safely, scalability to handle millions of peers, failure handling to provide the service durability, concurrency to allow multiple processes run at the same time, and transparency to enable everyone to benefit from any resource from any peer.

2.3 Structure of P2P Systems

The research for P2P systems in the literature can be categorized into two groups. The first area is building the P2P overlay network, and the second part is using the built-up P2P system in the most effective manner by utilizing the most effective algorithms to search or replicate the resources. The most important part in building a P2P system is the substrate. It has all the mechanism for managing the peers for joining, leaving and handling the failures. The files are managed by means of locating and storing them.

The P2P system architectures can be divided into two main categories according to the overlay construction, peer organization and management: structured and unstructured P2P systems. In structured P2P systems, the peers are organized to follow specific algorithms to locate and connect to a peer. On the other hand, in unstructured P2P systems, the peers have a certain degree of freedom compared with the ones in the structured networks, and they do not need to follow strict rules while forming the overlay networks.

The following subsections briefly describe these two types of P2P overlay networks with examples.

2.3.1 Structured P2P Systems

Since the peers are organized to follow specific algorithms to connect to the system, the structured P2P substrates has a mechanism to map the location of the peer or a file. The mechanism used by structured P2P system is called Distributed Hash Tables (DHTs) which build substrates with desirable scalability properties. All substrates using DHT use k-ary search which guarantees locating a file if it exists in the system in $O(\log_k(N))$ steps, where N is the number of files or peers in the system.

Chord [81], CAN [68], Pastry [69], and Tapestry [91] are famous DHTs introducing proprietary structures and algorithms for themselves for lookup, where to connect new arrival of a peer, and handling the leaves and failures. In the structured P2P systems the nodes are connected to each other in a certain predefined topology, e.g. to form tree, ring, hypercube, torus, d-dimensional cartesian space, mesh, and butterfly networks.

[31] and [73] examine several DHT organizations. [31] defines the flexibility as the freedom a peer has in choosing its neighbors and address the effect of peer failures, end-to-end path latency, and the locality of message exchange on the performance of structured P2P systems. [73] expresses the self organization as the most attractive property of the structured P2P systems.

Although the DHTs are very powerful to locate an existing content, it does not have a wide deployment application area such as Gnutella and Kazaa [34]. [34] summarizes the reasons of this in three items. (1) The queries shall be exact-match to locate the content, (2) peers are very transient with the time requirements of at least $O(\log(N))$ steps in every join/leave to restructure the system, and (3) in file-sharing applications, most of the queries are for the most popular files which removes the necessity for a DHT implementation.

2.3.2 Unstructured P2P Systems

In contrast to structured P2P networks, a peer can connect to any peer in the network and a file can be stored on any peer in the overlay network. A peer joins to the system by knowing an already participating peer and then uses a flooding algorithm to obtain knowledge about other participants. A peer has a freedom to connect to any peer without constructing a predefined geometric topology while forming the overlay network. The flexibility in the topology generation of the substrate enhances the resilience of the system by tolerating the transient behavior of peers. The absence of predefined geometric topology while forming the substrate also eliminates the need to restructure the overlay network in the arrival and departure of the peers.

The search mechanism used in structured P2P networks supports flexible queries to lookup for partial words. In the location of a peer or a file, a participant performs a query by asking some/all of his neighbors about a given query by the flooding algorithm. The process is iteratively applied by the neighbors until reaching the maximum number of iteration number to reduce the overhead. A search may fail to locate an existing file since the scope of the search is limited to a specified number of steps. The main drawbacks of unstructured P2P systems are: (1) the expensive search cost caused by flooding, (2) lack of guarantee on locating a requested item that exists in the system.

Gnutella [6], Napster [8], and Kazaa [7] are very popular examples on unstructured P2P substrates.

There are other studies classifying the P2P substrates into generations. [73] divides the substrates into three generations. The first generation is using centralized directory to index the contents of all subscribers as used in Napster [8]. The data exchange is performed in P2P methods while search is performed in centralized lookup service. The problem with this approach is the bottleneck in server bandwidth and performance requirements.

The second generation removes the requirements of the centralized directory service, and all services are performed in P2P approach. Gnutella [6] and Kazaa [7] are the most popular examples of this generation where the problem of this approach is longer queries since a peer queries its neighbors and they forward the queries to its own neighbors. Hybrid solutions to the P2P systems offered by several studies [35, 74] can also be classified under second generation.

Finally, [73] classifies the DHTs as the third generation of P2P systems. In his studies, the DHTs are investigated according to analysis and design perspectives. Since this generation fits the structured P2P approach above, it has the same advantages and disadvantages already mentioned.

2.4 Clustering Methods

The increasing demand in peer to peer networking motivated us to find new ways to improve performance and usage capabilities for user in peer to peer systems. The most critical performance bottleneck of a peer to peer system is the time spent for searching the content that is looked for. The key to improving the speed and efficiency of the information retrieval mechanism is to minimize the communication costs, that is, the number of messages sent between the peers, and to minimize the number of peers that are queried for each search request. Doing so reduces the aggregate load generated by each query across the network.

In [18, 23, 27, 34, 54, 61, 80, 85, 89, 90], clustering is used to bring: (1) a distributed control mechanism to enable the users to have a scalable and fault-tolerant system. (2) efficient data retrieval with less overhead traffic and a quick response for newly published resources. (3) a robust architecture and service which does not fail due to the failure of a small number of critical elements. (4) efficient resource usage. There are two main approaches in clustering. The first one is to use the location of the peers or the round trip time (RTT) between the peers in order to construct the clusters which localizes the traffic exchange. Although, it does not improve the hit ratio for any search, it reduces the cross traffic across different clusters or between remote peers. The second approach uses interests of the peers to assign them into clusters in which the peers are grouped according to their interest categories. The main advantage of interest based clustering is the reduction of average search depth which reduces the time for a search and reduces the flooding traffic. It also increases the hit ratio, since the peers are organized with respect to their interests and potential future requests. In both approaches, the peers are either grouped according to their interest categories or monitored in time to analyze their behaviors to construct clusters adaptively.

In the literature, there are different approaches to form interest and geographical location-time based clusterings. [90] uses dynamic landmark technology to construct the locality aware clustering which reduces the average distance of the constructed overlay network almost 9 times compared with traditional random structures. [54] constructs a hierarchy of clusters constructed by using the time metrics to reach the other peers. They try to use the RTT value of two peers as a metric to estimate the bandwidth between two peers.

[34] uses two levels of aggregation to divide peers into clusters. In the first level, peers attached to the same network are grouped into the same network-level cluster, which they call network cluster for short. The second level aggregates all networks within the same AS into a larger cluster, called AS cluster. Each cluster has a super peer, which performs special functions based on the cluster level. The routing tables of the routers running internet are used to divide the peers into clusters to build a topology similar to the topology of Internet. In addition to the two levels, a method is employed using bootstrap peers to be contacted by joining peers in order to direct the joining peers to the appropriate cluster. [54] uses similar approach dividing peers into clusters. Instead of using the network topology and routing table of the Internet routers, a message exchanging mechanism between peers to get information on the available bandwidth between peers is used. RTT is used as the main metric to disperse the peers into clusters.

[27] offers to use some special servers to keep the virtual topology of internet and try to estimate the coordinates of the peers by using the help of those topology servers called Tracers. [40] tries to estimate relative coordinates of the peers in internet by using RTT values from a node to the well known landmarks to estimate the inter-node distances. [61] offers to use coordinates based architecture to calculate the internet network distance of the peers which is called Global Network Positioning (GNP) to calculate absolute coordinates. [61] evaluates their method by comparing the results with two existing methods offered by [27, 40]. The results show that the methods offered by [61] and [27] outper-

forms the method offered by [40].

[85] offers layers of hierarchy while using interests (ICN) which is based on cache management. The higher level classes are connected by following a de Brujin graph. The lower level nodes are clustered by using their interests. By enabling the routing through higher level classes when necessary reduces the overhead traffic. ICN is self-organizing, fully distributed, scalable, and logically hierarchical. In ICN, the upper level is bound by de Bruijn graph. Nodes in the lower level self-cluster based on interest. [85] offers methods for constructing the hierarchy, rules for search, join and leave, and finally for cache management. In [80], interest based locality is used. The idea behind the study is: if a peer has a particular piece of content that one is interested in, then it's likely that it will have other pieces of content that one is also interested in. The method improves the performance of a search while it reduces the flooding traffic.

[18] proposes a peer to peer protocol in order to form self organizing topologies. The idea forming the overlay network is to connect to the peers who have already provided good files in the past, and potential new peers probably providing the required files. They also consider the trustworthiness of the overlay network by keeping local trust scores for the neighbor peers by examining the past behavior and contribution of the neighbors.

[89] worked out on topology aware P2P on demand video streaming. Their approach is designing a distributed content discovery protocol with organizing the peers into topology aware overlay networks. The approach they have is to benefit from caching the content very close to the edge of the network. It reduces the traversed path of the packets over the internet.

Most of the articles related to the clustering is concentrated on reducing the search time for an item, and improve the success rate of the search to be performed in order to reduce the need for external resources (i.e. CDN servers) rather than ordinary peers to guarantee finding the needed content in time. Besides, studies on video streaming show that one of the most important concepts is to localize the traffic in order to reduce the traffic exchange between the networks.

2.5 Streaming Architectures

The next subsections explains the studies on the literature about how a streaming system can be established. How a server based video streaming system works and the bottlenecks of the system and the potential solutions of such problem -P2P systems- has been examined. The P2P contribution is investigated as how to force the peers to contribute to the P2P system and how to solve the problem of resource locating in the absence of feeding peers.

2.5.1 Content Distribution Networks

The evolution in the telecommunication sector during the last decade enabled the service providers to develop applications requiring higher bandwidth such as audio/video streaming. The popularity and high interest in such applications bring several problems with them. In order to solve the performance and scalability problems caused by serving to the enormous number of potential users, third party systems are built to bring the delivery system close to the end systems. The systems established to deliver digital media conforming to the requirements of high bandwidth and faster delivery is called Content Distribution Network (CDN). The CDN systems follow similar structures followed by the client/server paradigm by utilizing the servers distributed at point of presence (POP) locations of the service providers. Akamai [1] and Digital Island [2] are the most famous CDN companies providing content delivery services including media streaming.

The key point of such systems is introducing local traffic exchange for [34]: (1) faster delivery of digital medium by using high speed local links, (2) less traffic overhead by keeping the content very close to the clients, (3) supporting higher quality by bringing less variable delay, and low loss rate. In the CDN systems the original copy of the media is kept either in the center of the content owner or the central server farm of the CDN system. In a CDN system, servers are deployed at the edge of the Internet, and clients request media streaming service from their closest CDN server. In a CDN architecture, a media data is first pushed to multiple CDN servers, each of which serves clients in its designated domain [20]. The CDN, system uses DNS redirection techniques to direct a client to the closest server to minimize the traffic path between the client and the CDN node that provides the service [11]. The main focus on CDN systems is to reduce the delays experienced by a user within the boundaries of human perception threshold.

The CDN systems employ several applications to monitor the traffic across the internet and to direct the clients to the appropriate servers especially using the information where they are connecting from.

For limited number of users, CDNs are perfect for delivering any kind of content, however, as the number of users becomes uncontrollable, the bandwidth capacity and computing power of CDNs cannot answer the requests properly even for poor quality multimedia streams [20, 34]. The main drawback of the CDN system is the scalability to reach to the millions of internet users. The biggest CDN operator Akamai has thousands of servers dispersed around the world. Akamai, with its tens of thousands of servers spread in an intelligent topology, still can't serve more than 300,000 concurrent streams, which is never going to impress the TV network used to audiences in the millions.

High costs to implement CDN servers and the limited scalability opportunities because of the huge amount of traffic requirements, implementing video services by using only CDN servers is not a feasible solution.

2.5.2 Peer-to-Peer Video Streaming Systems

Video-on-demand (VoD) has been one of the most popular applications of the Internet today especially with the creation of the video sharing website YouTube in 2005 by which users can upload, view, and share video clips [4]. Current YouTube video clips are relatively short; around 98% of the clips' lengths are within 10 minutes. Moreover, bit rates of YouTube clips are moderate; most video clips have a bit rate at around 330 Kbps with two other peaks at around 285 Kbps and 200 Kbps [16]. However, a demand for higher bit rate (potentially DVD quality) and longer videos (such as full length movies) is expected to arise in the near future [55]. For such applications, conventional infrastructure-based VoD systems fall short in meeting the scalability requirements that would arise. These scalability requirements stem from bandwidth and server hardware costs that are in direct proportion with the increasing demand. An alternative is to use Peer-to-Peer (P2P) architectures that were successfully used for file sharing, for VoD systems. The key to the success of P2P VoD architectures is the fact that when there are more users, there is not only more demand but also more capacity since each participating peer would also devote some of its own resources to the system making the system scale to meet the demands of future VoD systems. On the other hand, as indicated in the studies carried by [87], the characteristics of the networks show that the systems are heterogeneous and dynamic, and only 5% of the peers can act as servers [71]. This is because of poor availability, peer dynamicity, long startup time, and unwillingness of the peers to share their contents with the others. In addition, as summarized in [59] an effective P2P video streaming system should have appropriate video coding scheme, assignment of streaming rates and data segments to the sending peers to manage the peer heterogeneity and dynamicity, efficient overlay network construction, selection of best sending peers among the possible candidates, monitoring the characteristics of the underlying network to adapt into changing conditions dynamically, and incentives for participating peers. In order to compensate for the problems caused by the characteristics of P2P systems, we have to employ some intelligent algorithms. Those algorithms should be based on allocating enough number of seeding peers with enough computation power, bandwidth and source capacity.

P2P computing claims to address the problem of organizing large scale computational societies [50] with the objective as pooling and coordinating large amount of resources. Although the power of the P2P computing comes from sharing the resources among all internet users, it is very difficult to find volunteers to share their content with others, as this requires sharing of CPU and bandwidth. Especially, most of the P2P systems forces the users to share their resources amongst the others. [74] offers a method to force the subscribers to contribute to the system as a content provider to provide the content they have obtained from the system. Their approach is to guarantee enough number of seeds on the net in any interval. They either force the subscribers to provide some limited amount of seeds or to serve to unlimited customers within limited time. [44] offers to select the nodes with the larger available bandwidth (LBF) to get the maximum bandwidth available, and also offers to select the ones with minimum amount of bandwidth in order to have maximized number of feeding sources in order to reduce the effect of the failure in the contribution ratio. Lastly, to guarantee the duration of the feeds, they offer a method to select the servants from the nodes that are using some network resources. This idea is based on that no one would leave the system without getting the required service. They also offer a method to download the future parts of the content being streamed starting from the end to be kept in local storage to reduce the time to get the stream.

One of the proposed P2P architectures for video streaming relies on application-level multicast in which the peers assist the video server in forming the multicast tree and distributing the streaming media over the tree [51]. In this architecture, bandwidth costs are shared among the peers which reduces the burden on the server and also on the network in terms of bandwidth. Several systems for media streaming are proposed using the framework for application-level multicast such as the end system multicast (ESM) [17] and PeerStreaming [51]. However, since the leaf nodes of the multicast tree are not involved in carrying the data for others, this creates a fairness problem among peers. This observation has led to the construction of separate multicast trees so that a leaf node in one tree becomes an interior node in the other, and the content is sub-streamized and the sub-streams are distributed using separate trees with disjoint interior nodes, improving the fairness of the overall system [15].

In an alternative P2P architecture, the peers might be used as streaming servers directly. In such systems, in contrast with the application-layer multicast approach, a peer may serve a video even when it is not receiving the video from other peers, or when receiving a different video [78]. In a P2P streaming system, participating peers usually have different upload and download capacities, i.e., access capacity is asymmetric. For example, the upload capacity is 128-512 Kbps and download capacity is 512-2048 Kbps for a typical Asymmetric Digital Subscriber Line (ADSL) connection for residential Internet users. The asymmetric nature of the access bandwidth makes it problematical for high bandwidth video applications to be served by a single peer. In such scenarios, the video is encoded into multiple sub-streams each of which is served by a separate peer. This method not only reduces the required upload capacity for each pair but also balances the load among the peers. Throughout this thesis, we call such P2P systems multi-stream, which will be the focus of the current study. In multi-stream P2P VoD systems, upon a video request, the system attempts to set up a session during which multiple sub-streams from a number of peers would be sent towards the requesting user. A session would not be set up (the request would not be admitted) if the system cannot find the required number of peers having a copy of the requested video.

There are several studies considering peers heterogeneity into their designs, and addressing the problem of carefully selecting supplying peers based on the network conditions and peers characteristics. The video streaming framework offered by [62,63] provides a mechanism to combine multiple sender contribution with varying speed to a single receiver. The receiver assigns the packets/rate to each sender by the rate allocation algorithm in order to maximize the utilization and minimize the total packet loss. In [36, 38, 42] PROMISE, Gnustream and Collectcast are offered as the prototype applications to be used for P2P video streaming. They overcome the problems caused by the underlying P2P network dynamics, its heterogeneity, and peer dynamicity. They have the following functionalities: (1) selecting the best sending peers to recover from failure or degradation and dynamic switching of sending peers by periodically probing the peers, (2) monitoring the characteristics of the underlying network for peer failure or degradation detection and streaming quality maintenance, (3) assigning streaming rates and data segments to the sending peers by aggregating the bandwidth, (4) adaptively controlling the buffer to reduce the fluctuations in the streaming sessions.

Another key issue peculiar to P2P streaming systems is that the behavior of the peers are unpredictable, i.e., peer nodes churn. In these systems, peers are free to join the system randomly, get their requested content and then leave the system, which results in peer churn (or dynamicity). The churn rate is defined as the number of peers leaving the system in a given period and high churn rates in P2P VoD systems lead to poor availability and high service interruption probability. The reference [21] proposes two metrics to measure the dynamicity of the network called persistence and size change. The former measures the number of peers remaining the same between two consecutive snapshots of the network. The larger the persistence, the lower the dynamicity. The latter measures the fluctuations in the network size. The simulations show that the persistence is related with stability and by utilizing more stable nodes, one can have the P2P system more scalable.

[39, 48, 50, 70, 88] bring high level of accessibility of required resources distributed among the peers in P2P systems. [48, 50, 88] analytically determine the performance of the P2P systems by modeling a P2P network having N redundant copies of a requested content using exponential node failure and recovery processes. Finally, by using the equations obtained, the number of required redundant copies for a content or similarly number of nodes in a P2P system in order to guarantee the Fault Tolerance for a given amount of time is calculated. A stochastic fluid model is developed in [48] to study the characteristics and limitations of P2P streaming systems including the peers' realtime demand for content, peer churn, heterogeneous capacity of the peers, limited infrastructure capacity, and peer buffering and playback delay. The numerical evaluations of the model show that the performance of large systems is better than the performance of small systems since they are more resilient to bandwidth fluctuations caused by peer churn. A decentralized mesh-based layered video streaming overlay network architecture is proposed in [32] aiming robustness and resilience to high churn rate of peers. The architecture provides redundancy not only in the network paths, but also in the multi-layered video content.

2.5.3 Hybrid CDN-P2P Streaming System

The pure P2P streaming system would starve from the lack of available resources since one of the most important issues in P2P streaming system is to locate the seeding peers. In [92], a hybrid overlay network is constructed to benefit from the best features of tree topology and random dissemination mechanism, thus making effective use of the available bandwidth in the network. [74, 77] offer a method to combine CDNs and P2P systems to provide a hybrid solution to video streaming with unstructured P2P systems by using a CDN or a single supplying peer to be used together or alternatively which limits the major gain from using both systems together to bring the fault tolerance. The approach of [74] is based on using a CDN or a single supplying peer to be used together or alternatively which limits the major gain from using both systems together to bring the fault tolerance. Most of the studies on hybrid P2P streaming proposes and analyzes a media streaming system that integrates content distribution network (CDN) and P2P system. The idea is that CDN servers are used to initiate or 'jump-start" the system by streaming to a few requesting peers. As time goes on, peers will acquire sufficient capacity to serve other requesting peers, and hence free the CDN server. [20] tries to guarantee enough number of seeds on the network in any interval by using the CDN servers as actual streaming server during the initial stage and as an index server to keep the information of the peers. However, using such index server becomes the bottleneck of the system because of the scalability problems. They also force the subscribers to contribute to the system as a provider of the content they have obtained from the system.

Although the Hybrid CDN-P2P approach solves most of the problems to start the streaming sessions, after some period of time there would be no need for CDN server contribution within the system. However, as it can be seen from the study of [20, 74] the contribution of the CDN servers would be needed to guarantee the existence of all sources within the system forever. In [65], the CDN servers are kept actively in the system during all the streaming session to be used as the high available failover node to reconstruct the lost connections with the fastest mechanism. In the study, the proposed video streaming system has a design including CDN servers, contributing peers, combining multi-rate resource contribution in order to maximize the number of peers to be used as a serving node. A challenging method is offered to collect the different samples of a file from different contributing nodes (either CDN or P2P node) and fault tolerant system design to have perfect view of the file being streamed.

In addition to the network architecture, the coding technique of the content to be streamed is very effective on the performance of the streaming applications. Next section gives an idea on how to code the video to have better performance on streaming applications.

2.6 Video Coding for a P2P System

In multi-stream P2P VOD systems, packets may not be delivered to the destination due to churn or packet losses stemming from congestion in the network. To prevent service interruption in multi-stream P2P VoD systems, one needs a robust mechanism to deal with peer churn and network congestion. For example, when a supplier peer leaves the system and the sub-stream from this peer is therefore no more fed towards the destination, the mechanism needs to ensure minimal quality degradation until a replacement peer can be found. For this purpose, different multiple sub-stream encoding techniques are described in the literature [55]. In Forward Error Correction (FEC)-based methods, the media stream is broken into data units and a Reed-Solomon (RS) coding is applied to L consecutive data units to generate K redundant parity data units [51, 55]. The reception of L data units out of the $M = L + K$ data units will then be enough to recover the original stream in a system in which a sub-stream is formed from each of the M data units. However, it would also be desirable to devise a multi-stream coding technique where receiving a few sub-streams can lead to an acceptable quality, and receiving more streams can lead to better quality [55]. There are mainly two major video encoding methods for this purpose: Layered Coding (LC) [52] and Multiple Description Coding (MDC) [28]. In LC, a stream is broken up into M ordered sub-streams, the lowest layer being the base layer and the others enhancement layers, but the decoding of a sub-stream is possible only if all the sub-streams below are received correctly. Therefore, the quality of a media stream is not proportional with the amount of correctly received data in systems using LC. For example, in a multi-stream P2P system, if the base layer supplying peer unexpectedly leaves, this would lead to an interruption of the video playout at the receiver and all the information received up to that point would be useless until a new peer is found that can stream the base layer. This can only be avoided if each peer holds the base layer in addition to other layers [93]. In order to prevent this, FEC codes can be used with Scalable Coding (SC) [84]. On the other hand, MDC provides error resilience to packet losses by creating multiple sub-streams (or descriptions) that can be decoded independently. In MDC, the quality of the video would be enhanced as more descriptions are correctly received. FEC based robust source coding (MD-FEC) combines the advantages of LC and MDC by breaking the stream up into M sub-streams whereas correct reception of any L sub-streams guarantees the recovery of the lowest L layers [67]. In [9], a framework called Flexible MDC (F-MDC) is proposed to reproduce the descriptions by postprocessing already available video data on the fly by considering the number of available peers, and the current network conditions. F-MDC is used to develop a P2P streaming system aiming to improve the performance by adapting the number of base and enhancement layers, the redundancy and rate of each individual description, and rate allocation among the peers according to varying network conditions [9]. In [55], a new multi-stream encoding technique is proposed which eliminates the drawbacks of LC by giving equal importance to sub-streams and those of MDC by improving upon transmission redundancy.

Especially Fountain codes which are called as rateless codes are very suitable for increasing error resilience in a P2P system. An ideal fountain encoder can generate potentially infinitely many encoding symbols from the original data consisting of k symbols in linear time and receiver can reconstruct the original data from any k-element subset of received encoding packets in linear time. So if a peer leaves the system before the receiver receives k packets, the remaining packets can be received from another peer since any k-element subset is enough to reconstruct the original data.

Although, most of the contemporary studies use MDC alone or, use MDC with Solomon codes or tornado coding techniques, there is a need to build a new video coding scheme to be used for the streaming applications in P2P systems. In [76], a new multi-stream coding and transmission scheme, Redundancy Free Multiple Description (RFMD) Coding and Transmission is proposed that has been specifically designed for P2P VoD systems. All the substreams in RFMD have equal importance as opposed to the layered video coding. In the new coding scheme, only source bits contributing to reducing video distortion are transmitted by the supplying peers, so that there is no redundancy as is in conventional MDC. The simulation results in the study shows that the lack of redundancy in MDC significantly increase the streaming capacity of the P2P VoD system.

All the coding methods above can be used to deal with peer churn and network congestion. In this study, we do not emphasize a certain sub-stream encoding method but assume that for a high quality P2P VoD session to be maintained, at least a certain number of peers, say $L \geq 1$, need to have a copy of the requested video and be online and a connection needs to be established with each of them. One might optionally receive $K > 0$ extra (or redundant) sub-streams from K other peers to cope with peer churn and network congestion. Our scope in reliability modelling is peer churn but coping with congestion turns out to be a by-product. If the number of connections drops below L in the lifetime of the video then the quality of the VoD session is said to be unsatisfactory.

CHAPTER 3

RELIABILITY MODELING AND ANALYSIS OF P2P NETWORKS

This chapter presents the detailed modeling of P2P systems reliability analytically. In the analytical studies, a Markov chain–based reliability model for a P2P network having N redundant copies of a requested content using exponential node failure and recovery processes is offered. The model is also enhanced to include backup connections to improve the performance and reliability of the systems.

3.1 Introduction

In this chapter, we propose a simple Markov chain-based stochastic model to study multi-stream P2P VoD systems. We assume that N copies of the video are available at N peers in a P2P network. The parameter N is representative of the size of the network and the popularity of the content. We assume a fixed N and we deliberately avoid a seasonal non-stationary model for the number of peers in the network for the sake of simplicity. On the other hand, we use a simple Markovian model for peer churn. In this model, each peer is allowed to be in one of the two alternating states *ON* and *OFF*. A peer is active, i.e. can request and provide sub-streams in the *ON* state and it is idle in the *OFF* state. The amount of time a peer spends in each state is exponentially distributed with parameters μ and λ for the *ON* state and the *OFF* state, respectively, which is illustrated in Fig. 3.1. It should be clear that the Markov chain parameters λ and μ are representative of the dynamicity of the network. We assume that at least L sub-streams are to be received for a high quality streaming session. We then study two models, namely Models A and B. In Model A, we do not receive

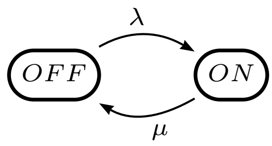


Figure 3.1: 2-state model of a peer.

extra sub-streams but once a peer departs, we immediately find a replacement (if available) in the system but once the number of online peers drops below L, the quality of the session is not acceptable any more. Model B is an extension to Model A in the sense that in reality when a peer departs from the system, it takes a while to find a replacement and set up a connection. In the meantime, the quality of the session degrades and key frames are lost possibly causing scene freezes, i.e., unacceptable quality. Therefore, one needs to use one of the above-mentioned sub-stream encoding mechanisms such as FEC or MD-FEC so that a receiver tries to maintain $M = L + K$ sub-streams (as opposed to L) to the extent possible. In this case, a peer departure would not jeopardize the quality of the streaming session unless the number of the remaining online connections drops below L. With appropriate sub-stream encoding mechanisms, it is clear that one can improve the churn resilience of the system. In this paper, we study Models A and B using the transient solutions of continuous-time Markov chains and in particular the existing theory of phase-type (PH-type) distributions. It should be noted that the scope of this paper is confined to the reliability analysis of P2P VoD systems. The procedures for selecting the best sending peers for each session, establishing connections and coordinating contributions from the peers are dealt with at the middleware layer, for example Collectcast [37], and are intentionally left outside the scope of this paper.

The remaining of this chapter is organized as follows. We provide a brief overview of phase-type distributions in Section 2. Section 3 describes the models A and B and certain performance measures are derived. Numerical results are given in Section 4. We conclude in the final section.

3.2 Phase-type Distribution

One of the tools that we use to study certain performance measures of the system is the well-known phase-type distribution (PH-type) which is defined as the distribution of time till absorption in a finite state continuous-time Markov chain [60]. For this purpose, we define a Markov process on the states $\{1, 2, \ldots, m, m+1\}$ with initial probability vector $\{v_1, v_2, \ldots, v_m, 0\}$ and infinitesimal generator

$$\mathbf{Q} = \begin{bmatrix} \mathbf{T} & \mathbf{T}^0 \\ \mathbf{0} & 0 \end{bmatrix}, \tag{3.1}$$

where $\mathbf{T}$ is an $m \times m$ matrix, $\mathbf{T}^0$ is $m \times 1$, and $\mathbf{Te} + \mathbf{T}^0 = \mathbf{0}$. Here, $\mathbf{e}$ denotes a column vector of ones of appropriate size. The matrix $\mathbf{T}$ is assumed to be nonsingular which ensures that the absorbing state can be reached from any other state [49]. We also define $\mathbf{v} = \{v_1, v_2, \ldots, v_m\}$. The last state $m+1$ is called the absorbing state. The time till absorption into this absorbing state is a random variable X which is said to have a PH-type distribution with representation $(\mathbf{v}, \mathbf{T})$. The distribution function of this process is then written as

$$F_X(t) = 1 - \mathbf{v}e^{\mathbf{T}t}\mathbf{e}, \;\; t \geq 0, \tag{3.2}$$

and its density function is then given by

$$f_X(t) = \mathbf{v}e^{\mathbf{T}t}\mathbf{T^0}, \;\; t \geq 0. \tag{3.3}$$

In many modeling applications including the one studied in the current paper, the absorbing state corresponds to a failure state and the random variable X should then be interpreted as the time to fail. The probability that the system has not failed yet at time t is called the success probability at time t and is denoted by $S(t)$ which is given by

$$S(t) = \mathbf{v}e^{\mathbf{T}t}\mathbf{e}, \;\; t \geq 0. \tag{3.4}$$

Finally, the moments of the PH-type distribution are given by

$$E[X^n] = (-1)^n n! \mathbf{v}\mathbf{T}^{-n}\mathbf{e}, \tag{3.5}$$

and in particular the Mean Time To Fail (MTTF) for the random variable X is written as

$$E[X] = -\mathbf{v}\mathbf{T}^{-1}\mathbf{e}. \tag{3.6}$$

3.3 Reliability Models

We describe and derive the relevant performance measures of interest for Models A and B below.

3.3.1 Reliability Model A

Model Description

In this model, we assume that there are N redundant copies of the video in the network, and L sub-streams are required for successful reception of video. The peers are assumed to leave and join back the P2P system according to the Markov model given in Fig. 3.1. In Model A, we do not receive extra sub-streams but once a peer departs, we immediately find a replacement if available in the system. However, when the number of online peers drops below L due to peer churn, we reach a failure state beyond which video reception is unsatisfactory. We construct the birth-death type continuous-time Markov chain in Fig. 3.2 to describe the number of online peers in this system. The transition rate from state i to $i+1$ is denoted by $\lambda_i = (N-i)\lambda$; and the rate from state i to $i-1$ is denoted by $\mu_i = i\mu$. For the Markov chain in Fig. 3.2, the steady state probability P_i of having i online peers in the steady-state is written by

$$P_i = \binom{N}{i} \frac{\rho^i}{(\rho+1)^N} \ , \tag{3.7}$$

where $\rho = \lambda/\mu$.

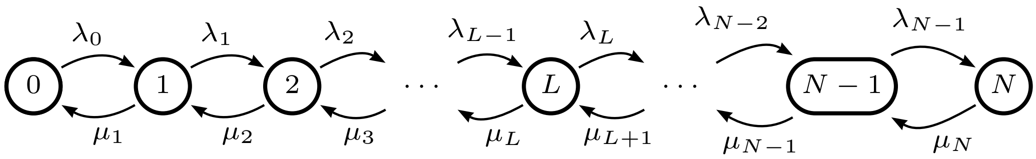


Figure 3.2: Markov chain used for Model A.

Admission Probability

For a new video request to successfully initiate a session, L peers need to be online in the system. Therefore, the probability that a video request would be admitted into the P2P system so as to initiate a session is denoted by P_{adm} which is given by

$$P_{adm} = P[i \geq L] = \frac{1}{(\rho+1)^N} \sum_{i=L}^{N} \binom{N}{i} \rho^i. \tag{3.8}$$

Success Probability

The success probability, denoted by $S(t)$, for a given video with length t, is the probability of successful reception throughout the entire duration of the video given that the VoD request is admitted. To find $S(t)$, we merge all states i such that $i < L$ into one absorbing failure state. It is not difficult to show that the distribution of time to fail is a PH-type distribution with representation $(\mathbf{v}, \mathbf{T})$ where

$$\mathbf{v} = \frac{1}{P_{adm}} \left(P_L, P_{L+1}, \dots, P_N \right),$$

and $\mathbf{T} =$

$$\begin{bmatrix} -(\lambda_L+\mu_L) & \lambda_L & & & \\ \mu_{L+1} & -(\lambda_{L+1}+\mu_{L+1}) & \lambda_{L+1} & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{N-1} & -(\lambda_{N-1}+\mu_{N-1}) & \lambda_{N-1} \\ & & & \mu_N & -\mu_N \end{bmatrix}.$$

One can then find the success probability $S(t)$ using the identity (3.4) as well as all the moments of the time to fail including the MTTF using (3.5).

Mean Time-to-fail (MTTF)

Although an expression for the MTTF is available through (3.6) for a video session that has just initiated, in this subsection we attempt to find a closed-form expression for the same quantity. For this purpose, note that once the session is initiated, it is maintained as long as the number of online peers $i \geq L$. So, the expected time till failure is the mean first passage time to state $L-1$.

Denoting the average time to transition from state i to state $L-1$ by T_i, we can write MTTF as

$$MTTF = \frac{1}{P_{adm}} \sum_{i=L}^{N} P_i T_i. \tag{3.9}$$

First consider state N. The mean time to transition from state N to state $L-1$ is equal to the sum of the mean time to transition from state N to state $N-1$ and the mean time to go from state $N-1$ to state $L-1$. Since the holding time at state n is exponentially distributed with mean $1/\mu_N$, T_N can be expressed as

$$T_N = \frac{1}{\mu_N} + T_{N-1}. \tag{3.10}$$

Now consider state $N-1$ for which we can write T_{N-1} as

$$\begin{aligned} T_{N-1} &= \frac{1}{\lambda_{N-1}+\mu_{N-1}} + \frac{\mu_{N-1}}{\lambda_{N-1}+\mu_{N-1}} T_{N-2} \\ &\quad + \frac{\lambda_{N-1}}{\lambda_{N-1}+\mu_{N-1}} T_N. \end{aligned} \tag{3.11}$$

Solving (3.10) and (3.11) simultaneously for T_{N-1} in terms of T_{N-2} yields

$$T_{N-1} = \frac{1}{\mu_{N-1}} + \frac{\lambda_{N-1}}{\mu_N \mu_{N-1}} + T_{N-2}. \tag{3.12}$$

Following the same steps for state $N-2$ gives the identity

$$\begin{aligned} T_{N-2} &= \frac{1}{\lambda_{N-2}+\mu_{N-2}} + \frac{\mu_{N-2}}{\lambda_{N-2}+\mu_{N-2}} T_{N-3} \\ &\quad + \frac{\lambda_{N-2}}{\lambda_{N-2}+\mu_{N-2}} T_{N-1}, \end{aligned} \tag{3.13}$$

which, when solved together with (3.12), produces

$$T_{N-2} = \frac{1}{\mu_{N-2}} + \frac{\lambda_{N-2}}{\mu_{N-1}\mu_{N-2}} + \frac{\lambda_{N-1}\lambda_{N-2}}{\mu_N \mu_{N-1} \mu_{N-2}} + T_{N-3}. \tag{3.14}$$

Repeating this procedure for all states $i \geq L$, we obtain the closed form expression for T_i as

$$T_i = T_{i-1} + \sum_{p=0}^{N-i} \frac{\prod_{j=i}^{i+p-1} \lambda_j}{\prod_{j=i}^{i+p} \mu_j}, \tag{3.15}$$

with $\prod_{j=i}^{i-1} \lambda_j$ should be equal to unity. Substituting $i = L$ in (3.15) and noting that $T_{L-1} = 0$, T_L is written as

$$T_L = \sum_{p=0}^{N-L} \frac{\prod_{j=L}^{L+p-1} \lambda_j}{\prod_{j=L}^{L+p} \mu_j}. \tag{3.16}$$

Back-substituting (3.16) in (3.15) with $i = L + 1$, we obtain T_{L-1} as

$$T_{L+1} = \sum_{p=0}^{N-L} \frac{\prod_{j=L}^{L+p-1} \lambda_j}{\prod_{j=L}^{L+p} \mu_j} + \sum_{p=0}^{N-L-1} \frac{\prod_{j=L+1}^{L+p} \lambda_j}{\prod_{j=L+1}^{L+p+1} \mu_j}. \tag{3.17}$$

Continuing back-substitution up to $i = N$ provides us the general closed form expression for T_i:

$$T_i = \sum_{k=L}^{i} \left[\sum_{p=0}^{N-k} \frac{\prod_{j=k}^{k+p-1} \lambda_j}{\prod_{j=k}^{k+p} \mu_{i-q+k}} \right], \quad L \leq i \leq N \tag{3.18}$$

Substituting (3.18) in (3.9) and recalling that $\lambda_i = (N-i)\lambda$, $\mu_i = i\mu$, the MTTF expression is finally written as

$$MTTF = \frac{1}{\mu P_{adm}} \sum_{i=L}^{N} \left(\binom{N}{i} \frac{\rho^i}{(\rho+1)^N} \sum_{k=L}^{i} \left(\frac{1}{k} \sum_{p=0}^{N-k} \rho^p \frac{\binom{N}{k+p}}{\binom{N}{k}} \right) \right) \tag{3.19}$$

The MTTF can also be calculated using the mean time to absorption of the PH-type distribution as in (3.6) through a matrix inversion. Our numerical evaluations reveal that both methods produce identical figures. One immediate implication of the MTTF expression (3.19) for Model A is that the MTTF is inversely proportional with μ as ρ is fixed or equivalently the MTTF decreases proportionally with increased dynamicity of participating peers.

3.3.2 Reliability Model B

Model Description

The shortcoming of Model A is that when a peer departs from the system, it actually takes a while to find a replacement and set up a connection. In the meantime, the quality of the session degrades. To make sure that this degradation does not lead to unsatisfactory quality, we use sub-stream encoding mechanisms with error control such as FEC or MD-FEC so that a receiver tries to maintain $M = L + K$ sub-streams (as opposed to L sub-streams of Model A). We call this model Model B. In this case, a peer departure would not jeopardize the quality of the streaming session unless the number of remaining online connections drops below L. The goal of this section is to model and quantify the impact of the parameter K on system resilience to churn.

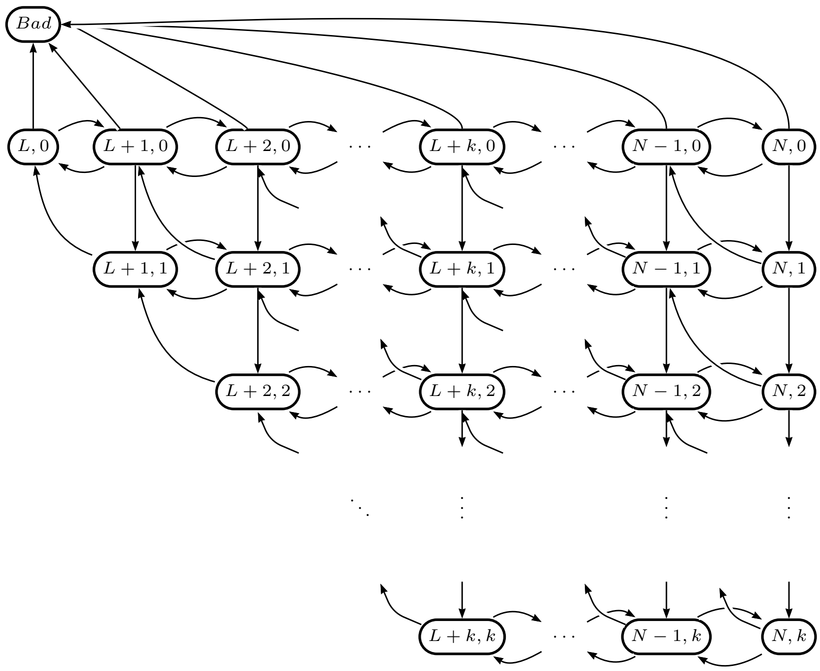


Figure 3.3: 2-dimensional Markov chain for Model B.

The P2P system with extra connections (sub-streams) for error resilience is modeled as a two-dimensional Markov chain on the state space $(i, j), L \leq i \leq N, 0 \leq j \leq K$, where i denotes the number of online peers and j denotes the number of extra sub-streams that the receiver is fed with. The structure of this Markov chain is given in Fig. 3.3. The absorbing state is called the *FAIL* state in which the instantaneous number of connections drops below L leading to unsatisfactory video decoding performance. We now describe the transitions among the states of this Markov chain. First, a transition to the *FAIL* state is possible from state $(i, 0)$ with rate $L\mu$ for each i. An east-transition is possible whenever $i < N$ with a rate of $(N - i)\lambda$. A west-transition is possible if there are online peers that the client is not connected to, i.e., $i > L + j$, with a rate of $(i - L - j)\mu$. A north-west transition occurs when a peer that the client is connected to goes idle and hence may always occur outside the $FAIL$ state with rate $(L+j)\mu$. Note that when $j = 0$, a north-west transition corresponds to the special case of a transition to the *FAIL* state. Finally, a south transition occurs when the client connects to an online peer as an extra connection. It is obvious that this type of transition is possible if there are online peers that the client is not connected to, i.e., $i > L+j$; and the number of back-up connections has not already reached its maximum which is K, i.e., $j < K$. This type of transition is assumed to occur with rate $(i - L - j)\bar{\lambda}$. This new parameter $\bar{\lambda}$ denotes the rate at which the client can connect to one of the available peers and is used for modeling the query time to find a peer that has the media and to setup the associated connection. However, there are $(i - L - j)$ such peers which leads us to the factor $(i - L - j)$ above. This observation stems from the fact that the minimum of n exponentially distributed random variables is also exponential with mean equal to the sum of the means of the n individual random variables.

Admission Probability

The probability of a session to be initiated for Model B depends on the policy for session initiation. As a P2P system policy, we require that at least $L + K$ nodes should be online for a video request to be admitted and a corresponding session to be initiated. Similar to the previous subsection, we write the admission probability as

$$P_{adm} = \sum_{i=L+K}^{N} P_i = \frac{1}{(\rho+1)^N} \sum_{i=L+K}^{N} \binom{N}{i} \rho^i. \tag{3.20}$$

Success Probability

Similar to Model A, we model the time to fail with a PH-type distribution with representation $(\mathbf{v}, \mathbf{T})$ that is governed this time by the Markov chain in Fig. 3.3 with the *FAIL* state absorbing. We enumerate the states of the Markov chain in Fig. 3.3 such that we first write the states for the first row from left to right and then the second row and so on. With this enumeration, we can write the vector $\mathbf{v}$ as

$$\mathbf{v} = \frac{1}{P_{adm}} \Big(P_{\{L,0\}}, \ldots, P_{\{N,0\}}, P_{\{L+1,1\}}, \ldots, P_{\{N,1\}}, P_{\{L+2,2\}}, \ldots, P_{\{N,K\}} \Big)$$

with $P_{\{i,j\}} = 0$ when $j \neq K$ and $P_{\{i,K\}} = P_i$ where P_i is given by (3.7). The matrix $\mathbf{T}$ can be formed using the descriptions of the transitions among the states in subsection 3.3.2. Finally, the success probability with respect to time (i.e., video length) and the MTTF can be calculated following the identities (3.4) and (3.6), respectively.

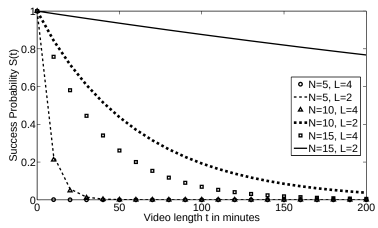
3.4 Numerical Experimentation

In this part we present the numerical experimentation for offered models: model A and model B based on the assumption that the ratio λ/μ is equal to 1, meaning that the online and offline periods of a peer are equal on average.

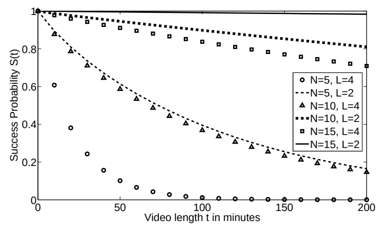
In order to investigate the behaviors of the peers with different λ/μ values, the numerical experimentation based on model B is repeated with $\mu/\lambda = 1/2$ to demonstrate the system characteristics when the average offline time of peers is twice of the average online time of the peers. The experimentation is repeated when the ratio $\mu/\lambda = 1/4$.

3.4.1 Model A

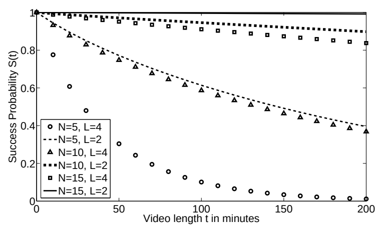
In our numerical experimentation, we assume that there will be no failure during the playout phase of a video and the typical peer capabilities over the Internet are assumed. We use $L = 2$ and 4 seeding peers with $N = 5, 10, 15$ peers having the required media, respectively. We study the cases corresponding to $1/\lambda = 1/\mu = 4$, 64, 128 minutes which are the average online and offline times for a peer in the system (the duration of online and offline period of a peer is assumed equal). The resulting success probabilities $S(t)$ are plotted in Fig. 3.4 as a function of the video length t. There are three parameters affecting the success probability of a session for Model A: the number of total seeding peers N, the number of connections L and the rate at which individual peers come online and go offline, i.e., λ and μ, respectively. We show that success probabilities increase with the number of total seeding peers N and decrease with increasing number of connections L. The reason for the latter behavior is that when L increases the contribution of a single peer to the overall video decreases, reducing the efficiency



(a) $1/\lambda = 1/\mu = 4$ minutes



(b) $1/\lambda = 1/\mu = 64$ minutes



(c) $1/\lambda = 1/\mu = 128$ minutes

Figure 3.4: Success probabilities with respect to time in Model A for $1/\lambda = 1/\mu = 4, 64$ and 128 minutes respectively

Table 3.1: MTTF Values for Model A in Minutes when $1/\lambda = 1/\mu = 64$ minutes

	$L = 2$	$L = 4$
$N = 5$	109.13	21.33
$N = 10$	969.74	101.46
$N = 15$	12092	594.40

of the system. On the other hand, the parameters λ and μ are representative of dynamicity of the system in the sense that the larger these parameters, the more dynamic the system would be. In dynamic systems, success probabilities are significantly reduced as demonstrated in Fig. 3.4. Similar conclusions can be drawn from Table 3.1 in which we tabulate the MTTF values for a number of scenarios. Based on the findings of Fig. 3.4 and Table 3.1, we are led to believe that aggressive swarming, i.e., large L, leads to an adverse effect on system reliability under Model A especially for small to moderate sized P2P systems.

If the success probability of a given video with length t in a certain network

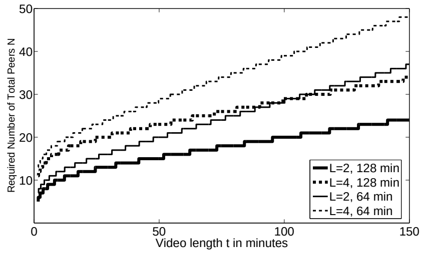


Figure 3.5: Required total number of peers for fault tolerance with respect to time (in minutes) for Model A. ($1/\lambda = 1/\mu = 64$, 128 minutes)

configuration is no less than 99%, we say the system is *fault tolerant* under that configuration for that video. Fig. 3.5 illustrates the required number of peers having the video content in order for the video session to be fault tolerant. We observe that the required number of peers is asymptotically linear in the video length. It is also clear that the network size requirements get stringent with increasing L and increasing dynamicity of the network.

3.4.2 Model B

In the evaluation of Model B, the success probability expression (3.4) and the MTTF expression (3.6) are used. During the evaluations, we use $L = 2$ and 4 seeding peers, $N = 5, 10$, and 15 peers having the required media, and the number of extra connections $0 \leq K \leq 3$, with $1/\bar{\lambda} = 2$ seconds, since typical average hit time in an unstructured peer-to-peer architecture is around 2 seconds [64]. This value is used throughout the evaluations in this paper. The cases corresponding to $1/\lambda = 1/\mu = 4, 64$, and 128 minutes are studied and the resulting success probabilities $S(t)$ are depicted in Fig. 3.6 as a function of the video length t, and the MTTF values are presented in Table 3.3. We have three main observations:

- When L is fixed, increasing the number of extra connections K also improves the reliability of the system in terms of $S(t)$ and MTTF but the marginal improvement observed by adding an extra connection, depends on the dynamicity of the system. For example, in the relatively static scenarios (scenarios (b) and (c) of Table 3.3), the gain in using $K \geq 2$

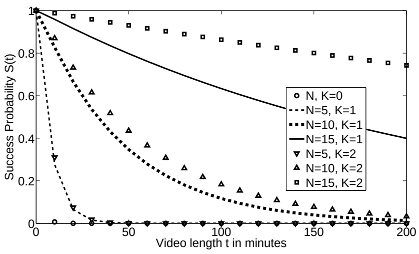
redundant connections is relatively small over the case when $K = 1$. However, for more dynamic scenarios (scenario (a) of Table 3.3), the choice of $K = 2$ presents a notable gain over the choice of $K = 1$.

- When K is fixed, the performance of the system worsens with increased L especially for small-sized networks, i.e., N is small. As before, increasing L reduces the contribution of each peer to the video decoding process, leading to reduced efficiency.

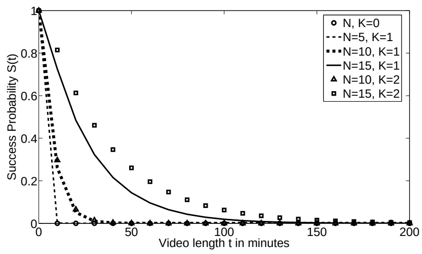
- The choice of $K = 0$ in Model B yields dramatically degraded performance when compared to Model A due to the nonzero time it takes to find a replacement peer. In Model B, when $K = 0$, the number of total peers N in the system does not affect the reliability of the system. This can be attributed to the fact that whenever N increases, although the number of available peers increases, the failure rate also increases proportionally since one can transition to the $FAIL$ state from all possible states.

We now present the the number of required peers N in order to provide 99% success probability for a given video length t in Fig. 3.7. It is clear that one can reduce the requirement on N by introducing redundancy in the form of extra connections or substreams.

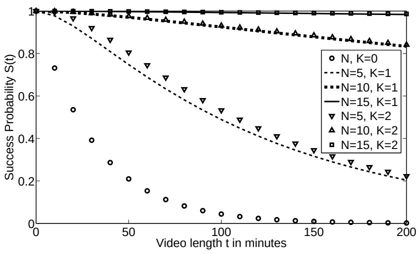
Considering the examples above, we are led to believe that maintaining more extra connections always improves the success probability of the system. Moreover, this improvement is magnified when the dynamicity of the P2P network increases. However, increasing K will also result in two impediments. Firstly, using more sub-streams increases bandwidth use and possibly leads to congestion in some network links. Secondly, the number of extra connections directly affects the admission probability P_{adm} of a VoD request. Since we have defined the success probability as the probability of successful reception throughout the entire duration of the video given that the session is admitted, we have failed to observe this effect until now. Therefore, we now define a new success probability $S'(t) = P_{adm}S(t)$ which is defined as the probability that a VoD request of length t is admitted and the video reception remains satisfactory throughout the entire session. Having defined the new success probability $S'(t)$ in this way, it is clear that indefinite growth of K will have an adverse effect on the success



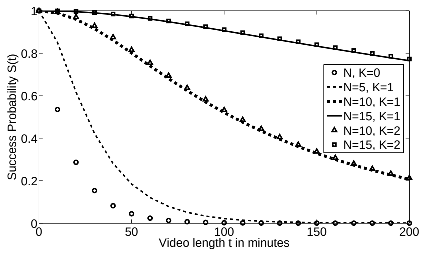
(a) $1/\lambda = 1/\mu = 4$ minutes, $L = 2$



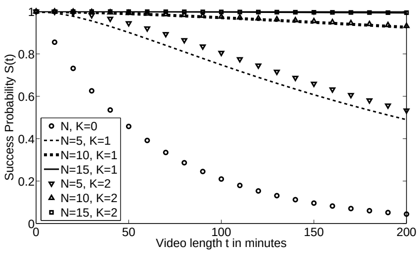
(b) $1/\lambda = 1/\mu = 4$ minutes, $L = 4$



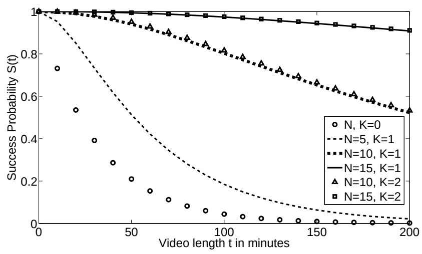
(c) $1/\lambda = 1/\mu = 64$ minutes, $L = 2$



(d) $1/\lambda = 1/\mu = 64$ minutes, $L = 4$



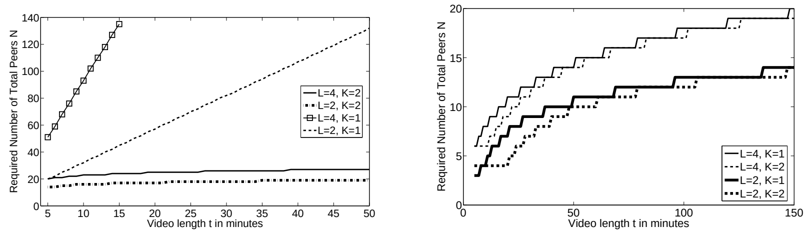
(e) $1/\lambda = 1/\mu = 128$ minutes, $L = 2$



(f) $1/\lambda = 1/\mu = 128$ minutes, $L = 4$

Figure 3.6: Success probabilities for Model B with respect to time. ($1/\lambda = 1/\mu = 4, 64$ and 128 minutes) and $L = 2$ and 4

probability since $S'(t)$ diminishes as K grows due to the reduction in the admission probability P_{adm}. As a result, we observe that the success probability $S'(t)$ starts decreasing beyond a certain value for K for a given video of length t in which case we denote this particular value for K by $K_1(t)$. Moreover, for relatively lower K values, the improvement in the success probability $S'(t)$ is slight beyond a certain threshold for K. We define a threshold value beyond which the success probability $S'(t)$improves no more than say 1% and denote

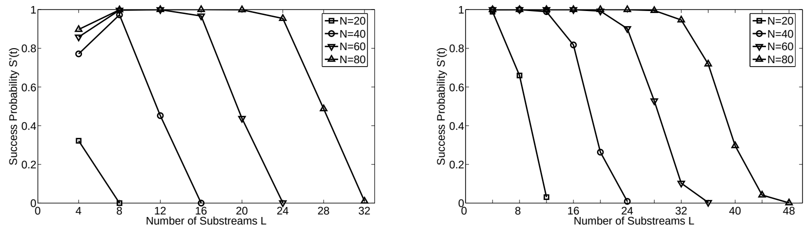


(a) $1/\lambda = 1/\mu = 4$ minutes (b) $1/\lambda = 1/\mu = 128$ minutes

Figure 3.7: Required total number of peers for fault tolerance in Model B for highly dynamic ($1/\lambda = 1/\mu = 4$ minutes) and relatively static ($1/\lambda = 1/\mu = 128$ minutes) networks

it by $K_2(t)$. The $K_1(t)$ and $K_2(t)$ values are given for $t = 90$ minutes in Table 3.2 for varying N and L. The value $K_2(t)$ should be viewed as a suboptimal choice for the number of extra connections (or redundancy) that we should have in the system since if the marginal gain of adding more redundancy in the system is minimal then we should probably be better off by not employing further redundancy since the less than 1% gain may not be worthwhile the increased bandwidth and resource utilization (which we do not model in this paper). We observe that for small-sized networks and increased dynamicity of peers, we should have more redundancy. Moreover, for small-sized networks, the $K_1(t)$ and $K_2(t)$ values tend to get closer.

We also study the streamization problem which is defined as finding the optimal number of sub-streams L for a fixed redundancy, i.e., $r = K/L$. In order to address this problem, we fix the redundancy to $r = 25\%$ and calculate the success probabilities $S'(t)$ for $t = 90$ minutes and for varying L. Since the redundancy is fixed to $r = 25\%$, we vary L such that $L = 4K$ for $K \geq 1$. The results are depicted in Fig. 3.8. We observe that there is a nontrivial optimal value for L above and below which we observe reduced performance for a fixed redundancy especially for small-sized networks and dynamic peers. On the other hand, the optimal value for L turns out to be the minimum feasible value for L ($L = 4$ for this example) for relatively static peers.



(a) $1/\lambda = 1/\mu = 4$ minutes (b) $1/\lambda = 1/\mu = 64$ minutes

Figure 3.8: The success probability $S'(t)$ as a function of the number of connections L.

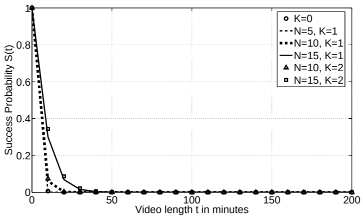
3.4.3 Model B with $\lambda/\mu = 2$ and $\lambda/\mu = 4$

In the evaluation of Model B with $\lambda/\mu = 2$ and $\lambda/\mu = 4$ (two times and four times longer average offline time than average online time) the success probability expression (3.4) and the MTTF expression (3.6) are used as in Model B. During the evaluations, we use $L = 4$ seeding peers, $N = 5, 10$, and 15 peers having the required media, and the number of extra connections $0 \leq K \leq 2$, with $1/\bar{\lambda} = 2$ seconds, since typical average hit time in an unstructured peer-to-peer architecture is around 2 seconds [64]. The cases corresponding to $1/\lambda = 4, 64$, and 128, $1/\mu = 8, 128, 256$ and $16, 256, 512$ minutes are studied and the resulting success probabilities $S(t)$ are depicted in Fig. 3.9 as a function of the video length t, and the MTTF values are presented in Table 3.4 and Table 3.5.

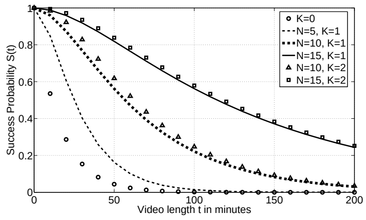
We have two main observations:

- When L is fixed, increasing the number of extra connections K also improves the reliability of the system in terms of $S(t)$ and MTTF but the marginal improvement observed by adding an extra connection, depends on the dynamicity of the system. For example, in the relatively static scenarios (scenarios (b), (c), (e) and (f) of Table 3.4) and Table 3.5, the gain in using $K \geq 2$ redundant connections is relatively small over the case when $K = 1$.

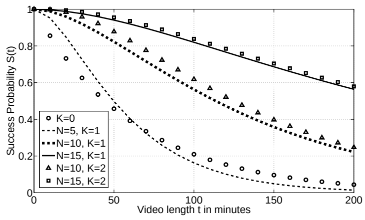
- Increasing the ratio λ/μ worsens the MTTF value with fixed L, K and N values since it results in having less resources in the system.



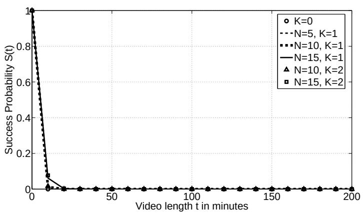
(a) $1/\lambda = 4$ and $1/\mu = 8$ minutes



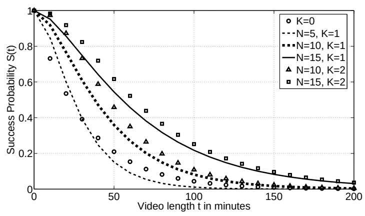
(b) $1/\lambda = 64$ and $1/\mu = 128$ minutes



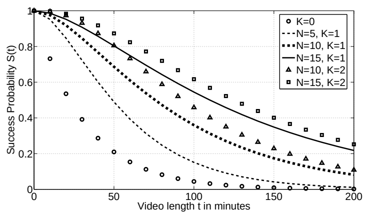
(c) $1/\lambda = 128$ and $1/\mu = 256$ minutes



(d) $1/\lambda = 4$ and $1/\mu = 16$ minutes



(e) $1/\lambda = 64$ and $1/\mu = 256$ minutes



(f) $1/\lambda = 128$ and $1/\mu = 512$ minutes

Figure 3.9: Success probabilities with respect to time.$\lambda/\mu = 2$, $\lambda/\mu = 4$ and $L = 4$

3.5 Conclusion

In this chapter, the reliability issue in a P2P system especially for video streaming applications is investigated. Unlike conventional video streaming systems

Table 3.2: The pair $(K_2(t), K_1(t))$ for $t = 90$ min. for varying N and L.

(a) $1/\lambda = 1/\mu = 4$ minutes

	$L = 2$	$L = 4$	$L = 6$	$L = 8$
$N = 20$	(2, 3)	(3, 3)	(3, 3)	(3, 4)
$N = 40$	(2, 4)	(2, 4)	(2, 5)	(3, 6)
$N = 60$	(2, 5)	(2, 5)	(2, 6)	(2, 7)
$N = 80$	(2, 4)	(2, 7)	(2, 5)	(2, 6)

(b) $1/\lambda = 1/\mu = 64$ minutes

	$L = 2$	$L = 4$	$L = 6$	$L = 8$
$N = 20$	(1, 2)	(1, 2)	(1, 2)	(2, 2)
$N = 40$	(1, 3)	(1, 3)	(1, 3)	(1, 3)
$N = 60$	(1, 3)	(1, 3)	(1, 4)	(1, 5)
$N = 80$	(1, 5)	(1, 3)	(1, 5)	(1, 3)

built around the client-server architecture, individual nodes in a P2P streaming system are far more likely to fail due to the contribution of all nodes in the system. Using a Markov chain model, we evaluated the architecture's reliability and studied its sensitivity to the nodes' average online and offline times. With the conservatively estimated parameters, we found that one can achieve system reliability surpassing dedicated conventional video streaming systems. This result is encouraging as the amount of nodes required having the requested content is very small when compared with the millions of peers over the Internet. Finally, the effect of having one and two extra connections is analyzed. The results indicate that, having one extra connection improves the performance aggressively. However, having more than one improves the performance further only in very dynamic networks. The model proposed in this paper provides the capability of computing the optimal value of the number of extra connections for given network parameters. Increasing the ratio of average offline time to average online time means having less resources in the system worsens results in the same environment.

Table 3.3: MTTF Values for Model B in Minutes.

(a) $1/\lambda = 1/\mu = 4$ minutes

(L, K)	$N = 5$	$N = 10$	$N = 15$
$2, 0$	2	2	2
$2, 1$	7.93	47.25	218.01
$2, 2$	8.67	59.67	667.81
$2, 3$	9.21	60.42	693.54
$4, 0$	1	1	1
$4, 1$	1.99	7.70	26.72
$4, 2$	–	8.43	37.85
$4, 3$	–	8.65	38.40

(b) $1/\lambda = 1/\mu = 64$ minutes

(L, K)	$N = 5$	$N = 10$	$N = 15$
$2, 0$	32	32	32
$2, 1$	131.11	980.31	10458
$2, 2$	140.67	1003	12060
$2, 3$	149.2	1009.9	12065
$4, 0$	16	16	16
$4, 1$	31.99	135.41	624.37
$4, 2$	–	138.09	643.78
$4, 3$	–	141.38	644.93

(c) $1/\lambda = 1/\mu = 128$ minutes

(L, K)	$N = 5$	$N = 10$	$N = 15$
$2, 0$	64	64	64
$2, 1$	262.52	1981	22460
$2, 2$	281.47	2009	24190
$2, 3$	298.54	2022.9	24198
$4, 0$	32	32	32
$4, 1$	63.99	271.76	1269.5
$4, 2$	–	276.38	1289.8
$4, 3$	–	282.95	1292

Table 3.4: MTTF Values for Model B in Minutes when the ratio for $\lambda/\mu = 2$

(a) $1/\lambda = 4$ and $1/\mu = 8$ minutes

(L, K)	$N = 5$	$N = 10$	$N = 15$
4, 0	1	1	1
4, 1	1.90	4.22	8.41
4, 2	–	4.69	9.30

(b) $1/\lambda = 64$ and $1/\mu = 128$ minutes

(L, K)	$N = 5$	$N = 10$	$N = 15$
4, 0	16	16	16
4, 1	30.4	70.23	147.8.37
4, 2	–	75.81	151.77

(c) $1/\lambda = 128$ and $1/\mu = 256$ minutes

(L, K)	$N = 5$	$N = 10$	$N = 15$
4, 0	32	32	32
4, 1	60.80	140.65	296.61
4, 2	–	151.68	303.73

Table 3.5: MTTF Values for Model B in Minutes when the ratio for $\lambda/\mu = 4$.

(d) $1/\lambda = 4$ and $1/\mu = 16$ minutes

(L,K)	$N=5$	$N=10$	$N=15$
$4,0$	1	1	1
$4,1$	1.85	2.86	4.15
$4,2$	–	3.42	4.72

(e) $1/\lambda = 64$ and $1/\mu = 256$ minutes

(L,K)	$N=5$	$N=10$	$N=15$
$4,0$	16	16	16
$4,1$	29.60	46.62	68.93
$4,2$	–	54.95	76.26

(f) $1/\lambda = 64$ and $1/\mu = 256$ minutes

(L,K)	$N=5$	$N=10$	$N=15$
$4,0$	32	32	32
$4,1$	59.20	93.30	138.05
$4,2$	–	109.92	152.56

CHAPTER 4

PROTOCOL AND CONNECTIVITY BASED OVERLAY LEVEL CAPACITY CALCULATION OF P2P NETWORKS

This chapter presents the detailed design of channel capacity calculation metric for the P2P architectures. The metric gives a result which is more effective than the current metrics such as query hit time, success ratio for P2P networks, *turnover* and *size-change*. The proposed metric is based on Shannon's L-channel capacity calculation idea to calculate the maximum rate of information (in bits per second) that can be transmitted over P2P network. The metric is calculated by monitoring the architecture and regularly getting the snapshots for the connections amongst the peers.

4.1 Introduction

Distributed nature of today's computing systems like Peer-to-Peer (P2P), grid, ubiquitous and multiagent systems triggered a paradigm shift in programming effort from the design of stand-alone, centralized computing bodies to the design of mostly primitive entities and well-designed interactions among them. As a consequence, protocol design became one of the most critical parts of the development of such systems when we consider the system's run-time efficiency, self organization ability and fault tolerance. Therefore, measurement of design quality of protocols is becoming an important issue. However, any quality judgment about such systems based only on given protocol specification can be misleading due to the lack of other important descriptive run-time dimension: evolving peer connectivity structure. Therefore, protocol-based system quality measurement requires the consideration of system's run-time dynamics.

Some of the most common P2P system performance evaluation metrics are; success ratio to locate the content after initiating a search, the time to locate the content and the traffic overhead introduced by the system, which are all related to the run-time performance. A system with high hit ratio, small hit time and localized traffic is treated as a good system. Those common metrics are not protocol-aware design level metrics but they can only judge about consequences of the run-time behavior of the P2P system. Although their calculation depends on peer connectivity characteristics, organization of the overlay network, and behavior of the individual peers; they do not give us any quantitative values about information flow capacity of the overlay network under study.

In this chapter, we proposed a metric for P2P networks based on Shannon's L-channel capacity calculation idea [75]. The metric calculates the maximum rate of information (in bits per second) that can be transmitted over P2P network (a.k.a. combinatorial capacity) caused by protocol and overlay-level connectivity. The metric considers both static protocol-level design time characteristics and dynamically changing overlay-level run-time peer connectivity structure. These properties of the metric make it more powerful and applicable to different protocols and peer connectivity snapshots. In the experimental work, the metric is first applied to the Gnutella 0.6 protocol for which message traffic explosion and search efficiency is a known problem. Then different from our previous work [66], it is applied to geographical-time, interest based, and two-way hybrid clustered version of the Gnutella network to obtain various peer conductivities. As indicated in [64] geographical-time based clustering algorithm enables us to exchange data with very high speed connections. Interest based clustering architecture allows a peer to locate a searched content in very few steps of a search. The two-way/hybrid clustering architecture has the superiority of both architectures. Since different clustering architectures have different connectivity structures which results in different capabilities and characteristics according to the organizations, different organizational structures are used to evaluate the current metrics and the one that we propose. For the evaluation, $number\ of\ query\ hits$, $unit\ query - hit\ response\ time$, $turnover$ [10] and $size\ change$ [21] metrics are used.

4.2 P2P metrics

In the literature there exist several studies to find out a good metric to measure the different characteristics of structured and unstructured P2P systems. The scope of them especially includes measuring the effect of dynamicity and churn; evaluating the trust to the content provided by other peers; enforcing the peers to contribute to the system by measuring their contributions and finally the capacity of the architecture to allow the peers to benefit from.

Since the behaviors of the peers are unpredictable in a P2P system, the effect of churn and dynamicity are addressed by several studies. [53] argued that traditional performance measures are uninformative for a continually running P2P system. They investigate the bandwidth consumption by a peer in the management of P2P networks and offered *rate* as the proper performance measure for the bandwidth consumption for maintenance purpose. $Half - life$ concept is offered as the metric to measure the dynamicity which is the total time taken for half of the nodes to change in the network. [10] offered *turnover* as another metric measuring the number of peers joining and leaving the network in the unit time. [21] offered two additional metrics to measure the dynamicity of the network called $persistence$ and *size-change*. The former measures the number of peers remaining the same between two consecutive snapshots of the network. The latter measures the fluctuations in the network size. Their simulations show that the persistence is related with the stability. By utilizing more stable nodes, one can have the P2P system more scalable.

Trusting to the content in a P2P system is another important concern. The past behavior of the peers are evaluated to form their reputation as a basis for a trust management system. The tradeoff in such evaluation is the behavior change of peers in time. [58] and [86] investigated the behavior of malicious peers which build good reputation by behaving normally at the beginning, and changing their behavior to cheating suddenly. They also mentioned the oscillation between normal and malicious behavior for peers. [22] pointed out the requirements on the dynamics of trust in P2P systems and proposed a trust metric which satisfies relevant requirements. The metric they offered is capable of detecting and penalizing the sudden changes and their potential oscillatory malicious behavior. In addition, it is flexible enough to handle different types of trust dynamics.

Forcing peers to adapt their behaviors to different protocols in order to maximize system utilization is an important topic of research. [82] offered a cooperation mechanism for individual peers to maximize the utilization of the system. The metric they offered for this purpose is based on the evaluation of the number of remote queries processed by each peer. They tried to maximize the remote work in the entire network by controlling the rate of query injection done at each node.

To the best of our knowledge in literature, there is no similar metric that calculates protocol and connectivity based information capacity of P2P systems at overlay level. Related to our study, in [48] a simple stochastic fluid model that seeks to reflect fundamental characteristics and limitations of P2P streaming systems has been worked on. However, the work is focused on modeling and they did not propose any flow based capacity calculating metric.

4.3 Discrete Noiseless Channel Model of P2P System

In the following, we will first give the definitions related to the discrete noiseless channel concept [75] and [43] and after each definition we explain its semantics in P2P context. The concepts of discrete noiseless channel and Shannon language have also been referred in [72], in the context of modeling of component-based systems.

Definition 1. Let A be a finite alphabet of symbols and each symbol say $a \in A$ has a possibly different duration in time defined by function $\tau(a)$. A *discrete noiseless channel* is a channel that allows the noiseless transmission of symbols based on their durations.

In our case, the P2P network is modeled as a noiseless channel and discrete alphabet symbols are the letters representing different type of messages of the given protocol. For example, for Gnutella-based protocols they are interpreted as i_{xy} (for ping), o_{xy} (for pong), u_{xy} (for query) and h_{xy} (for query hit) messages sent from peer x to peer y. Duration of each letter corresponds to the existence time or delay for the message type sent from peer x to peer y.

Definition 2. A finite string of say k letters from a finite alphabet say A, is called a *word* α of length k and its duration is defined to be the sum of durations of its individual letters (i.e. $\tau(\alpha) = \tau(a_1) + \tau(a_2) + \cdots + \tau(a_k)$).

A word of length k defined over message alphabet A is a sequence of message types where each symbol a_i is sent from one peer to another. Note that here we assume that a peer does not send message to itself, directly. However, a message sent by a peer to another may trigger the generation of new message(s) whose receiver can be the original sender peer. The total delay generated by a word of length k is the sum of unit delays generated by its individual messages.

Definition 3. Any set of possible words that can be constructed by using symbols from A is called *language L* defined over A. The channel that allows noiseless transmission of words only from the language L is called *L-channel*.

Definition 4. Let G be a directed graph whose edges are labeled with letters from the alphabet A. The corresponding *Shannon language* is defined by the set of words formed by following the edge labels on possible paths of the graph G.

Any P2P setup with its peer connectivity and protocol defines a Shannon Language. Consider a hypothetical P2P network snapshot represented in directed graph form in Figure 4.1. Nodes in the figure are peers. The edges define snapshot peer conductivities and they are labeled with the messages of the adopted Gnutella protocol. It is clear that the capacity of any P2P setup is not only dependent on the peer connectivity and the number of different types of messages but also to the message durations. In the example Figure 4.1, the edge values for different message types of the same peer pairs are supposed to be the same (e.g. $i_{12}o_{12}u_{12}h_{12} = 2$). On the other hand, they take different values (e.g. 3, 4, 5) for different peer pairs.

In Gnutella protocol, self ping, self pong, self query and self query-hit messages are not meaningful therefore the graphs of Gnutella based Shannon Languages do not hold any direct self-looping nodes. Also, for any neighbor peers (i, j), there are always eight edges due to symmetric nature of the protocol (i.e.

four from i to j and four from j to i). An example word of the language in Figure 4.1 can be $i_{12}i_{23}o_{32}o_{21}$. Note that in our modeling there is no assumption about run-time letter (or message types) orderings making the words of the language. The edges of the graph do not show a run-time message type sending scenario but the potential of sending a message type from one peer to another at any time.

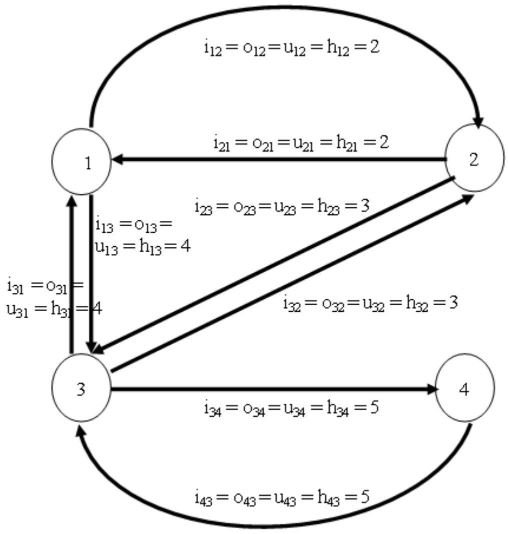


Figure 4.1: An example Gnutella based P2P network setup describing a Shannon language

4.4 The proposed metric

In the following subsection, we will give formal description of the proposed metric and then make a quality evaluation.

4.4.1 The description of the proposed metric

Definition 5. Given a Shannon language L, its *L-channel combinatorial capacity* is defined as

$$C_{comb} = \lim_{t \to \infty} \sup \frac{1}{t} \log (N(t)) \tag{4.1}$$

where $N(t)$ is the number of words in L for which duration $\tau(\alpha) = t$.

In Shanon's original work, a simple algebraic method for computing the combinatorial capacity of an L-channel has been given.

Definition 6. A *directed graph* G is defined by set of vertices $V = v_1, \ldots, v_M$ and by set of branches $B = b_1, \ldots, b_N$. Let init(b) and fin(b) be the initial and final vertices v and w of a branch b, respectively. Then the set $B_{v,w}$ of branches is defined to be

$$B_{v,w} = \{b \in B : init(b) = v,\ fin(b) = w\}$$

For the example in Figure 4.1, we can write $B_{1,1} = \emptyset$, $B_{1,2} = \{i_{12}, o_{12}, u_{12}, h_{12}\}$, $B_{1,4} = \emptyset$ or, $B_{3,2} = \{i_{32}, o_{32}, u_{32}, h_{32}\}$ etc. In fact, for our Gnutella based p2p setup we can say that $B_{i,i} = \emptyset$ for any $1 \leq i \leq M$ where M is the number of peers in the system at a given instant. Note that M may change during system evolution.

Definition 7. Given two vertices v and w from G, a *path* P of length k from v to w is a sequence of k branches $P = b_1 b_2 \cdots b_k$ with $init(b_1) = v$, $fin(b_1) = init(b_2)$, $\ldots$, $fin(b_{k-1}) = init(b_k)$, $fin(b_k) = w$. Also, we say that $len(P) = k$, $init(P) = v$, and $fin(P) = w$.

Then, the set of paths of length k from v to w is denoted by:

$$B^k_{v,w} = \{P : len(P) = k,\ init(P) = v,\ fin(P) = w\}$$

Using a directed graph G and an alphabet A, we can generate a language by labeling each branch of G with an element of A.

Definition 8. Let λ be a mapping (also called labeling) from B to A. The λ is called *right-resolving* labeling if and only if for each vertex v, the labels on all branches with init(b) = v are distinct.

It is easy to see that any graph labeling of Gnutella based Shannon Languages is right-resolving because of distinct labeling l_{xy} of any edge between peer x and y (see Figure 4.1). The label of a path $P = b_1 b_2 \cdots b_k$ is the concatenation of its branch labels: $\lambda(P) = \lambda(b_1)\lambda(b_2)\cdots\lambda(b_k)$. Therefore, $\lambda(P)$ defines a word over A whose duration is $\tau(\lambda(P))$ or simply $\tau(P)$.

Definition 9. The Shannon language generated by G with labeling λ is denoted by $L_{G,\lambda}$ and it is the set of all possible path labels P.

Given $L_{G,\lambda}$ one can compute the combinatorial capacity using partition functions [43].

Definition 10. Let s be nonnegative real number, and for a given pair of vertices (v, w), *branch duration partition function* is defined as:

$$P_{v,w}(s) = \sum_{b \in B_{v,w}} e^{-s\tau(b)} \tag{4.2}$$

The functions $P_{v,w}(s)$ can be thought of as entries in an $M \times M$ matrix $P(s)$.

Example 1. *For the graph of Figure 4.1, the obtained matrix is:*

$$\begin{pmatrix} 0 & 4e^{-2x} & 4e^{-4x} & 0 \\ 4e^{-2x} & 0 & 4e^{-3x} & 0 \\ 4e^{-4x} & 4e^{-3x} & 0 & 4e^{-5x} \\ 0 & 0 & 4e^{-5x} & 0 \end{pmatrix}$$

The spectral radius *(the magnitude of the largest eigenvalue) of the matrix $P(s)$ is represented by $\rho(s)$ and it is also called the* partition function *for the language $L_{G,\lambda}$.*

Theorem 1. *The combinatorial capacity of the $L_{G,\lambda}$ language is given by*

$$C_{comb} = \ln(s_0)$$

where s_0 is the unique solution to the equation $\rho(s) = 1$.

The combinatorial capacity for the above example can easily be calculated as $C_{comb} = 0,775$.

For the original proof of Theorem 1 the reader is referred to [75] and [43].

4.4.2 Evaluation of the metric

Before giving the experimental comparisons, we first make a verbal evaluation of the proposed metric in terms of the constraints of a P2P network in this subsection.

In general any metric used to evaluate a P2P system should

- *consider major P2P system primitives like delay, number of peer connections and their run-time changes during operation*: The duration of existence of a message type in channel corresponds to the delay concept in P2P systems. The proposed metric exploits not only the number of peer connections but also the peer connection topology. The delay associated with a message type is taken to be constant during system evolution and the peer connectivity may change independently.

- *consider time-varying interests and owned resources (like content) of peers*: The proposed metric does not directly consider time-varying peer interests and owned resources. However, interest and content based clustered versions of Gnutella network are worked on during experimentation. By this way, we obtain different connectivity structures handled by the metric.

- *give an idea about the number of messages send during operation and the amount of traffic being generated*: The number of messages sent, their intensivity and finding an upper bound for the generated traffic are the main concerns of the proposed metric.

- *reflect the run-time performance of the system*: The proposed metric does not directly intend to measure the run-time performance of the system. However, high information flow capacity of the system can be expected to result in better task (or query) execution performance. From this perspective, it reflects the run-time performance of the system.

- *be applicable to different types of P2P networks and consequently protocols*: The proposed metric supports any type of P2P networks and their protocols once their message types together with peer to peer channel delays are decided on.

4.5 Experimental Setup, Results and Discussions

4.5.1 The setup

In our setup, we have worked on a small group of peers (27 in number) with a certain initial join sequence modeled by uniformly distributed random arrival of new peers [80]. The P2P system consists of a group of peer nodes, each of which has different characteristics defined as a Gnutella overlay network. The system defines a bootstrapping node to provide functional overlay network behavior to the system. In all configurations a 3-node transit-stub network topology generated with GT-ITM [63] is used as the backbone. Nine leaf nodes are attached (1 ultrapeer and 8 leaf peers) to each stub node. The latency and the connection link for the backbone are configured similar to the real world instances. During the experiments the real world activities for the peers like join, leave, request a query, serve a query, forward the queries or block the queries have been modeled. The connection matrix for the system is taken by connecting to each peer and printing their connection list on each snapshot for each simulation.

The experiments are carried out in four different configurations: In the first simulation the pure Gnutella overlay network with leaf/ultrapeer structure is established to be used as a reference point for the second part of the simulations. The rest of the simulations are carried on the overlay networks generated by using geographical location-time, interest and two-way/hybrid clustered architectures on gnutella network. Clustering the overlay network utilizes different connectivity structures resulting in different capabilities and characteristics which enable us to evaluate the newly proposed metric based on the connectivity and organization of the peers in an overlay network. The second simulation batch focuses on the implementation of the time based clustering method which clusters the peers by the networks they are connecting from. The third simulation focuses on the implementation of the interest based clustering method which clusters the peers by the interest they have. The last simulation is for the implementation of Two-way/hybrid clustering architecture offered by [64]. It is aimed to figure out the effect of clustering on the system's combinatorial channel capacity calculation. The results give idea on the data transmission capabilities of the architecture evaluated.

In the Gnutella network the overhead of the ping, pong, query and query-hit messages are almost identical and the transmission delay generated by sending those packets are also negligible when compared with the queuing delay and propagation delay generated by the network when using high speed connections.

Thus, the delays (i.e. $\tau(a)$ values) for all message types listed are assumed to be equal. The duration for the simulation is set to 20 minutes. During the simulations, the connectivity matrix of the system is taken as a snapshot in predefined times starting at 100 seconds and in each 5 minutes consecutively. Since the P2P system studied in all configurations are distributed network, it was very hard to get the snapshot of the system from a central point. The connection matrix for the system is taken by connecting to each peer and printing their connection list on each snapshot for each simulation. Combinatorial capacity values have been calculated by using symbolic processing toolbox of MATLAB software Ver. 7.2.0 [41].

The same simulations are applied to all architectures again to calculate the *turnover* and *size change* values. The arrival and departure of the peers, and the total number of peers in the system is monitored and the status of the system is printed out in every 4 minutes. The duration of the simulation is set to 24 minutes.

In order to improve the reliability of the experiments, all the experiments are carried out 5 times, and the average of these are presented for as the outcome for each architecture.

4.5.2 Results and Discussions

Table 4.1 gives the comparison of different architectures in terms of the maximum rate of information flow (in bits per second) that can be transmitted over P2P network with protocol and overlay-level connectivity. In the table, the results from random gnutella architecture, geographical location-time based clustered architecture, interest based clustered architecture, Two-way/hybrid clustered architecture combining both approaches and the maximum channel capacity that can be obtained from the overlay network is summarized. The flow rate is calculated at 100^{th}, 400^{th}, 700^{th} *and* 1000^{th} seconds, respectively. The simulations are done three times for each type, and then the average values of the results are obtained. Although the channel capacities for all architectures are similar in the first interval, the capacity for pure Gnutella network and interest based clustering decreases from time to time while it stands almost the same level for the offered two-way/hybrid clustering model and the time based clustering. The last row of the table is the maximum channel capacity that can be achieved theoretically for the simulated overlay network. In this calculation, we used full mesh connection in order to see the upper bound for the channel capacity.

Table 4.1: Average information flow capacities calculated for the pure Gnutella system, for the Gnutella system with different clustering mechanisms and for the fully connected mesh system in different time periods. The measurements are taken at 100th, 400th,700th and 1000th seconds respectively.

Type of architecture	100 sec.	400 sec.	700 sec.	1000 sec.
Pure Gnutella Arch.	0.92	0.52	0.57	0.48
Time Based Clustered Arch.	0.94	0.86	0.89	0.89
Interest Based Clustered Arch.	0.95	0.70	0.72	0.66
Two-way hybrid Clustered Arch.	1.07	1.13	1.04	0.95
Max Capacity	1.47	1.47	1.47	1.47

The results summarized in Table 4.1 are compatible with and reflects the offered architectures' expected behaviors. For example, small delay (as in geographical location-time based clustering) known to improve the capabilities and the exchange traffic capacity in an overlay network. The small search time or search efficiency reduces the lookup time and improves the number of jobs can be handled in unit time. Finally, the two-way hybrid architecture combining both approaches is known to show better run-time performance than any of the single architecture, on the average. This evidence is also supported by the higher capacity results obtained for the two-way hybrid approach as a natural consequence. The upper bound values for static fully-connected mesh topology is given on the last line. Informally, its value can be treated as a bound for potential performance of the system under consideration. However, it should be clear that this conclusion is drawn through empirical observations without provision of any mathematical proof.

In table 4.2, turnover values are calculated for random Gnutella, geographical location-time based clustered, interest based clustered and two-way/hybrid clustered architectures. All architectures give almost similar results. In fact, turnovers as expected, do not directly give us an idea about the flow capacity of different overlay networks since it does not consider dynamically changing peer connection topology. As stated in the appropriate section the proposed metric gives an idea about the dynamicity of the network.

Table 4.2: Turnover values (arrival/departure) calculated for the pure Gnutella system and for the Gnutella system with different clustering mechanisms in different time periods. The measurements are taken cumulatively in every 240 seconds respectively.

Type of Arch.	240 sec.	480 sec.	720 sec.	960 sec.	1200 sec.	1440 sec.
Gnutella	(9,3)	(5,3)	(4,3)	(3,2)	(2,3)	(3,2)
Time Based	(9,3)	(7,4)	(6,4)	(3,2)	(3,5)	(4,3)
Interest Based	(10,3)	(3,3)	(6,3)	(6,4)	(5,6)	(4,4)
Two-way/hybrid	(10,2)	(6,3)	(7,4)	(5,4)	(4,5)	(5,5)

In figure 4.2, the size change in the different overlay networks are investigated. The figure shows all arrivals and departures to and from the system. The size for clustered overlay network has a trend to keep more peers in the network in the same conditions. When compared with the combinatorial capacity metric offered in this study, bigger the size results bigger capacity. Although, the number of peers in different architectures almost have similar number of peers in the overlay network, they have different combinatorial capacities since the combinatorial capacity of an overlay network is also effected by the number of connections among the peers. As we observed, higher the number of connections, bigger the capacity is.

Finally, a drawback of the proposed metric is its calculation inefficiency because of its intensive computation requirement. An attack for this problem can be to investigate the applicability of more efficient algorithmic approaches including parallel and/or distributed calculation of the metric.

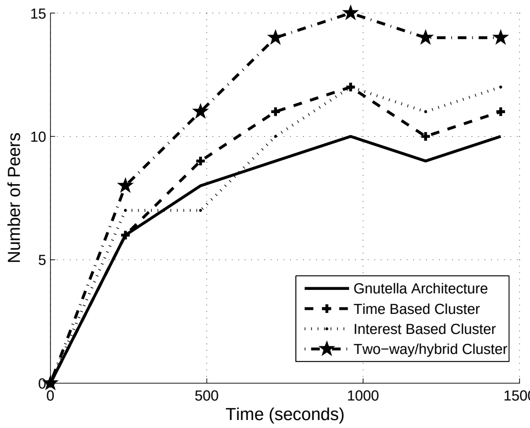


Figure 4.2: Size change for the pure Gnutella system and different clustering mechanisms

4.6 Conclusions

A metric for P2P networks based on Shannon's L-channel capacity calculation idea is proposed. The metric calculates the maximum rate of information (in bits per second) that can be transmitted over P2P network caused by protocol and overlay-level connectivity. Metric is applied to pure Gnutella network and its geographical location-time, interest based, and two-way/hybrid clustered versions. For comparative study purpose, the metric is compared with frequently used primitive P2P metrics including number of query hits, unit query-hit response time. In addition, non primitive P2P metrics: turnover and size change reflecting system dynamicity and churn are also considered. The obtained capacity results show that, the bigger combinatorial capacity is, the better the utilization of the same communication medium. The architecture having greater channel capacity gives better average hit time and the number of messages exchanged in the P2P system. The decreasing combinatorial capacity results in longer search results and hit time, and less number of queries send and hits in the system. The two-way/hybrid clustering we offered in [64] is observed to offer the highest channel capacity thus the highest hit ratio and low hit time.

CHAPTER 5

TWO-WAY/HYBRID CLUSTERING ARCHITECTURE AND HYBRID FAULT TOLERANT VIDEO STREAMING SYSTEM

This chapter presents the proposed fault tolerant hybrid P2P-CDN video streaming architecture. The two-way/hybrid clustering method is also proposed in order to increase the search efficiency and reduce packet retrieval time of the P2P system. The simulation results show that the proposed system increases the availability of the streams with the best quality while keeping the load of the CDN servers in required limits.

5.1 Introduction

In this chapter, a novel hybrid video streaming scheme is proposed which integrates peer to peer video delivery systems into dedicated CDNs together with fault tolerant schemes. The offered method solves the scalability issues of the CDN systems and the unreliability of P2P systems. In order to evaluate the performance, in addition to the average hit time and hit ratio for searched content, the channel capacity metric proposed in [66] is used. This metric calculates the maximum rate of information (in bits per second) that can be transmitted over P2P network (a.k.a. combinatorial capacity) caused by protocol and overlay-level connectivity.

Although, the analytical studies [48, 50, 88] show that fault tolerance can be obtained by only using pure P2P systems, the simulation studies show that the required quality of service (QoS) cannot be obtained from pure P2P systems. Thus, the proposed system uses the CDN servers to bring fault tolerance by using them as a backup system to locate the required segments of the files while searching for other peers to switch from CDN server in order to keep the load of the CDN servers below specified level thresholds. By this way, the CDN servers are used to provide the initial seeds and as the high available failover node to reconstruct the lost connections with the fastest mechanism. The difference of the method proposed from current state of the art studies is, the CDN servers are not used as index servers for the system as offered by [20] in order not to lead to a high load caused by the operation. Additionally, the offered method uses the contribution of the CDNs not only for initial seeding stage to feed the system with the initial streams but also during all the time. However, we only get a few segments of a failed node which is not available prior to the playout phase in order to guarantee the QoS. The cost of the method is very limited since the CDN servers are directly replaced by a new servant after completing the search on contrary with the offered method by [74].

The second contribution of this study is the increase in the search efficiency by offering a new clustering method. Although the currently used clustering methods limit the use of CDN servers, the offered Two-Way/Hyrid clustering technique outperforms all the other clustering offerings, by bringing efficient data retrieval with less overhead traffic and fast resource locating and efficient resource usage. In the model, the clustering methods are used to coordinate and manage the overlays created by CDNs or P2P contributors to reduce the traffic in the system.

5.2 Hybrid P2P System Architecture

In the following subsections the main components of the proposed system; the offered Hybrid Fault Tolerant Streaming (HFTS) architecture will be given in detail in order to solve already discussed problems. Then, the effect of different clustering methods including Two-Way/Hybrid clustering method on the system and the overload incurred by applying HFTS is explained in detail.

5.2.1 Hybrid fault tolerant streaming system

Even implementing the most intelligent algorithm to build a P2P structure does not solve the quality degradation in server–less video streaming. In order to guarantee the required Quality of Service (QoS) in video streaming we still need the help of CDN servers. The HFTS system is offered to bring all the required functionalities together for efficient and high quality P2P video streaming system.

The reliability model discussed in [48, 50, 88] show that if we have enough number of peers connected to the system, locating data existing in the system is guaranteed. The proposed video streaming system has a design including CDN servers and contributing peers, combining multi-rate resource contribution in order to maximize the number of peers to be used as a serving node. A new method is developed to collect different samples of a file from different contributing nodes (either CDN or P2P node) and a fault tolerant system is offered to have perfect view of the file being streamed. The offered model

- uses a dispersion algorithm to organize all the peers to benefit from the reception of search results in the fastest means and reduce the traffic between the networks.
- uses CDNs as a backup peer which do serve all the peers. If the content scheduled for a peer has not been received before the deadline to safely play the content (the deadline is calculated as the playing time minus the time to request&receive the content from the CDN), the source is set as failed. When a problem happens with a source, the client tries to locate a new one within a limited time. If it cannot succeed, the CDN starts to feed the client instead of the failing peer. While the streaming process continues, the client searches for other source peers to get the content provided by CDN. When a candidate is located, the feed from CDN is replaced by the new source.

HFTS manages the peers in the P2P overlay and the servers in the CDNs scattered throughout the entire Internet. There would be tens of locations (point of presence POP) connected to the internet to serve to the customers. In a typical design the CDN systems are organized by enabling the peer to locate the nearest CDN POP to reduce the network overhead and access time. The flat structure of the CDN is constructed by considering the Autonomous Systems and routing over Internet.

In order to maximize the utilization of the proposed system, contributions with different rates should be accepted to the system. To manage the contributions with different rates, a mechanism similar to Weighted Fair Queuing [56] is utilized in reverse approach in which all the contributing nodes have different weights with respect to their contribution speeds. All the sources are combined together to construct a single video stream to be played out.

In a typical P2P streaming system, the CDN or some dedicated contributing peers are used as an initial content provider at time t0. After enough seeds have been pushed to the system, the system uses only the contributing peers as the resource providers. In such systems there would be more than one contributing peer to provide the sustained rate for video stream. As one can see from Figure 5.1, any contributing peer can contribute to the system with a different speed. The offered method uses all the benefits of the multisource contribution even to benefit from the peers connected with slow dial up links.

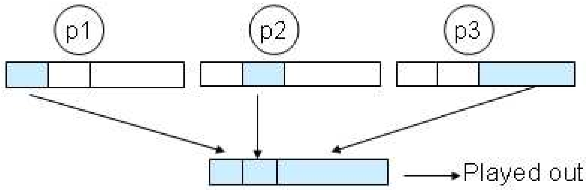


Figure 5.1: Combining the seeds from different sources into a single video stream

In HFTS, there are different types of peers. A peer would only request services from other peers (Prr - freeloaders) or a peer would either use or provide services to others (Prs - servants). Only Prs type of peers can be used as a resource provider within the P2P system. A contributing peer can use network services (Prsa active user) or can be idle or looking for a service (Prsp idle contributor). In HFTS architecture, all contributing nodes are selected with some specific selection criteria. In order to guarantee the durability of the service that will be used, the strategic nodes (Prsa) are selected. If there is no Prsa type of peer having the content that are looked for, Prsp type of nodes can be used as a contributing node. The motivation behind the selection criteria is that no one would leave the system without getting the required service. In order to solve the problems of locating a contributing peer within the boundaries of the jitter buffer and the unavailability of a contributing node with the required content, CDN servers are used to provide unavailable content within the system as in Figure 5.2 which gives a basic explanation on the CDN contribution of the offered system.

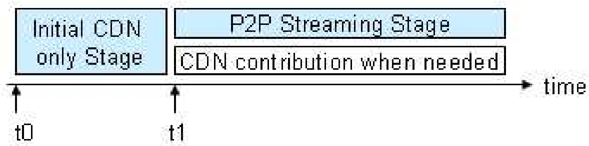


Figure 5.2: CDN contribution to guarantee reliability of a video stream

In the P2P video streaming system, the time to locate another serving peer (either Prsa or Prsp) would be longer than tolerable by the consumer, some degradation in service quality would be encountered. The HFTS system offers a method of grouping the whole streaming content into categories with respect to the serving nodes. Assuming that there are N supporting active peers, the system divides the whole content into N sub resources ranging RSi to RSn with respect to the contributing peers. Then it informs the CDN server about the parts of each contributing peer. If there happens to be a connection loss with any contributing peer, the system switches its state to CDN connection while searching for a new peer or peers to provide the content that is already being served by CDN server.

Our approach also controls the load of network traffic on the CDN POP. In the algorithm when there is a need for a packet the nearest POP is located by considering not only the congestion to reach the POP but also the channel capacity to reach to the POP to minimize the delivery of the requested packet.

The operation of HFTS algorithm can be summarized as follows where Bs is the total streaming bandwidth, Bp is the bandwidth used from peers, Bc is the bandwidth used from CDN servers, RSi is the resource provided by a peer, Prr is a freeloader, Prs is a servant, Prsa is an active user, and Prsp is an idle contributor.

```
(Start the search from the network to locate peers)&&
        (Start using the services from CDN servers)
if (Bp <= Bs) {
   Use the resources from peers (Bp) and
   the resources from CDN (Bc = Bs - Bp)
   to fill the streaming services
} else {
      if (Bp > Bs) {
         if (∑BPrsa > Bs)
            Select Prsa nodes
         if (any Prsa fails) {
            (use CDN for RSi) &&
            (search for Prsa node to replace CDN)
            Locate the nearest CDN with tolerable load
         } else {
            Select all Prsa nodes + some Prsp nodes
            to fill up Bs
            if (any Prsa || Prsp fails)
               (use CDN for RSi) && search for Prsa ||
               Prsp node to replace CDN
               Locate the nearest CDN with tolerable load
        }
    }
}
```

The offered method can be configured in different ways. The system would redirect a query to a CDN server in the failure of any contributing node or it can send a query to the P2P system and wait for tolerable amount of time. The time to wait would be longer than the average hit time plus some safety margin to reduce the traffic in the network. The waiting time before switching can be calculated adaptively with an algorithm similar to Exponential Weighted Moving Average [56] method which is used in calculating the timeout value for TCP packets. However, using the limits of the playback buffer set during the experiments is preferred.

The final part of the HFTS is to provide the coordination between CDNs and peers. When a connection fails with a source peer during a playback session, the system locates the nearest CDN server and supports the stream by providing the required blocks. One of the main contributions of this study is the continuous availability of CDN peers. The current P2P CDN approaches either have limited CDN contribution at the startup or within some specified time during the session [20, 74].

Even though the HFTS solves the problems in locating a resource provider peer within a specified time by using the CDN servers, the search efficiency can further be increased by using clustering methods.

5.2.2 The use of Clustering on HFTS

As stated before, several studies exist in the literature for P2P clustering. As stated previously, there are two types of clustering approaches in the literature: location-time based and interest based. The main advantage of location-time based clustering scheme is localizing the traffic exchange. Although, it does not improve the hit ratio for any search, it reduces the cross traffic across different clusters or between remote peers. On the other hand, the second approach disperses the peers with respect to the interests. The main advantage of this type of clustering is the reduction of average search depth which reduces the time for a search. It also increases the hit ratio, since you organize the peers with respect to their interests and potential future requests. Based on the analysis about topology and interest aware clustering to reduce startup latency and service interruption probability in [64], the proposed Two-Way/Hybrid clustering which is the combination of location and interest based clustering is selected to use with the HFTS system. The method considers the interests of the peers at the beginning to form the interest clusters while starting to investigate the behaviors of the peers to adaptively organize them to more dynamic overlay networks. In parallel, the peers are clustered in location based manner by where they are connecting from. This index is also dynamically updated to enlarge the scope of the local network into somewhat bigger local networks by evaluating the RTT values of the neighbor peers in close networks. The results in the study indicate that; by employing Two-way/hybrid clustering, we can improve the probability of finding a resource when needed; even the time to find the content is reduced. The method also tries to use the nearest content available, resulting in the reduction in delivery time and network load.

In the clustered model, all components of the CDN system is organized by a geographical location-time based clustering algorithm to locate the nearest (in time) CDN POPs to reduce the network overhead and access time. It has a dynamic structure to adapt itself to the changing conditions of Internet links.

Another overlay is constructed for P2P individuals. All the peers are clustered together to locate not only the nearest (in time) but also the most suitable peer (or the group of peers) first and use the services provided among themselves. All peers are organized with respect to their connection location. Therefore, the network they are connecting is the starting point of the clustering process. The results of the queries are ordered by the connection distance. The peers with small delay and enough bandwidth are selected to build a group of feeding peers to minimize the network traffic. In addition to the geographical location-time based clustering the peers, interest based and two-way/hybrid clustering methods in different steps of the simulations has been applied. The two-way/hybrid clustering method will be explained in detail in the following subsection.

5.2.3 Two-Way/Hybrid Interest and Time based Clustering

In the proposed technique, the currently motivated clustering techniques time-based clustering and interest based clustering techniques are combined into a single clustering method which will benefit from interest based clustering while considering the time to reach the contents. The idea behind of this clustering technique is (1) almost guarantee finding a searched content, (2) retrieve the content from the node having the least cost. The proposed method executes two different clustering strategies in parallel to divide peers into clusters. Figure 5.3 shows the basic components of the proposed system including peers, hybrid clustering layout and the traffic flow amongst the system components.

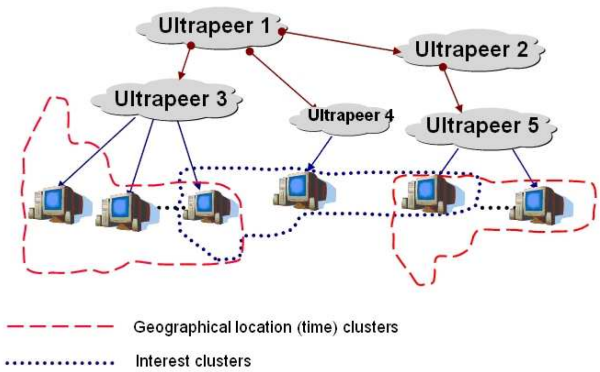


Figure 5.3: A typical Hybrid clustered overlay network combining two different cluster method into a single clustering mechanism.

All the peers are clustered together to locate not only the nearest (in time) but also the most suitable peer (or the group of peers) first and use the services provided among themselves. All peers are not only organized with respect to their connection locations but also clustered with respect to their interests. Therefore, the network they are connecting from and their interests are the starting point of the clustering process. The results of the queries are ordered by the connection distance and channel capacity. The peers with small delay and enough bandwidth are selected to build a group of feeding peers to minimize the network traffic and latency. Although, the traffic is tried to be localized by selecting the nearest peer, in the case of inexistence of the searched content in the same location based clustered peers, the peers in the same interest cluster helps to locate the required content easily.

The method allows any type of peers to contribute in any speed, to benefit from any resource in the internet. The proposed clustering method offers a method of grouping the whole streaming content into categories with respect to the serving nodes. Assuming that there are n supporting active peers, the system divides the whole content into n sub resources ranging from P_i to P_n

in the set of all peers P with respect to the contributing peers. F_i is the i^{th} segment of the whole video file F, $L_i(t)$ is a sorted sequence of required segments to be downloaded from peer i in the geographical location-time based cluster at time t, $G_i(t)$ is the interest cluster counterpart of $L_i(t)$, $A(t)$ is the collection of $L_i(t)$ and $G_i(t)$ ($A(t) = L_i(t) \cup G_i(t)$) which is the complete set of frames conforming the requirements of F_i with the links to source peers to download.

1. When a peer has a request from the network it initiates a search to locate the supplying peers for each partition of the requested content.

2. After initiating the query, the responses to the queries have been collected by the query initiator. By analyzing the replies, the peer prepares the plan $A(t)$ through searching the content F on the P2P network by considering peer contribution, connection and channel capacity characteristics.
   For $i = 1$ to length(P) do

   (a) Form $L_i(t)$ and $G_i(t)$

   (b) Construct $A(t)$ using $L_i(t)$ and $G_i(t)$ by preferring the items in $L(i)$ over $G(i)$

3. For each peer i in the plan $A(t)$

   (a) Send the request plan $L_i(t)$ and $G_i(t)$ by considering the supplying peers

4. For $j = 1$ to length(F) do {

   (a) Find peer i holding F_j in the plan $A(t)$

   (b) Download&Play F_j from peer i

   (c) For specified time intervals, try to update the plan $A(t)$ to replace the frames from $G(i)$ by potential peers contributing to $L(i)$.

   }

One of the main drawbacks of the offered system would be the overhead of dispersing the peers by using geographical location-time and interest based two-way/hybrid clustering. The overhead caused by including two-way clustering is not more than the cost caused by any ordinary peer clustering during the dispersion phase and control exchange during the execution. The heart beat to/from a peer is negligible even it is doubled in our algorithm since they are only communicating with the ultrapeers when compared with the heart beat traffic of ultrapeers. The overhead of ultrapeers and the core cluster heads on top of the system is also not more than a typical clustering system. The only additional overhead is the number of messages exchanged in the network, since there is a two-way/hybrid clustering algorithm. The overhead calculation of the two-way/hybrid clustering algorithm will be given in detail in the next section.

5.3 Numerical Evaluation and Experimental Results

5.3.1 The overhead incurred by HFTS and Two-Way/Hybrid Clustering

The overhead caused by including CDN servers is not different from the cost caused by any ordinary peer during the dispersion phase and control exchange during the execution. The heart beat to/from a peer is negligible since the size of the clusters is very small when compared with the heart beat traffic of CDN servers. The overhead of ultrapeers and the core cluster heads on top of the system is not more than any CDN peers in the system.

In order to give an idea about the overhead of HFTS, we considered the worst case (the most heavily congested part of the system CDN nodes) scenario in the network. In the simulations, the messages are exchanged in every 60 seconds. There are 900 peers and two CDN servers in the simulation environments. All peers have access to all CDN servers. The heart beat message from an ordinary peer to a CDN server is: 44 bytes (20 bytes for IP + 20 bytes for TCP + 4 bytes for data). Therefore, the inbound traffic overhead is (900 x 44 bytes x 8 bits/bytes) / 60 seconds = 5.28 Kbps. Every incoming heart beat is acknowledged (since it is TCP) using a message size of 40 bytes. Then the total outgoing heart beat traffic becomes: (900 x 40 bytes x 8 bits/bytes) / 60 seconds = 4.8 Kbps. Compared with the connection speeds for such systems, the overhead generated by HFTS is negligible.

In the simulations including two-way/hybrid clustering method, the messages are exchanged every 60 seconds. There are 800 Leaf peers and 100 ultrapeers in the simulation environments. An ultrapeer can handle atmost 16 connections. The heart beat message from an ordinary peer to an ultrapeer is: 44 bytes (20 bytes for IP + 20 bytes for TCP + 4 bytes for data). Therefore, the inbound traffic overhead is (16 x 44 bytes) / 60 seconds = 0.093 Kbps. Every incoming heart beat is acknowledged (since it is TCP) using a message size of 40 bytes. Then, the total outgoing heart beat traffic becomes: (16 x 40 bytes) / 60 seconds = 0.085 Kbps. Compared with the connection speeds for such systems, the overhead generated by Two-Way/Hybrid Clustering method is negligible.

Finally, by running both algorithms (HFTS and Two-Way/Hybrid clustering methods) will double the traffic on average in the CDN servers which increases the inbound overhead traffic to 10.56 Kbps and outbound traffic to 9.6 Kbps.

5.3.2 Simulation Settings

Video Data

The modeled video files in the simulation has the length of 90 seconds each and coded in Common Image Format (CIF) resolution. Each movie is coded into a base and four enhancement layers using H.264/Scalable Video Coder (SVC) (JSVM 4.0) [5]. To obtain each layer a combined spatial, temporal and quality scalability is used. Even though the number of enhancement layers can be increased, we keep it as four since in the simulations, the minimum number of peers connected is set to four. The average bitrate of a video (Rt) is set to be 1 Mbps.

Network Model

In our simulations, a heterogeneous system is assumed in which each peer has an uplink bandwidth of one of the following four values (64 kbps, 128 kbps, 256 Kbps or 512 Kbps). Certain number of peers (900) are defined with a certain initial join sequence modeled by uniformly distributed random arrival of new peers [80]. The P2P system consists of a group of peer nodes, each of which has different characteristics defined as in Gnutella overlay network. The system defines a bootstrapping node to provide functional overlay network behavior to the system. Each node in the network alternates between online or offline status. The connected time for a peer is modeled as an exponentially distributed random variable with mean λ. Similarly, the disconnected time is another exponentially distributed random variable with mean μ. Then the probability that a node is in online status is $\lambda \, /(\lambda + \mu)$. 100 popular movies with the characteristics of long term popularity models presented in [30] have been modeled. Each video has the same size but not the same popularity, with the popularity of these videos following the Zipf distribution [30] with parameter $\alpha = 1.1$. The Zipf́ Law holds that the number of requests to a file, and the number of different files at those types, have an inverse linear relationship.

When a node requests a video, it tries to find M serving nodes that have the M substreams of the video, with each node also having sufficient uplink bandwidth to serve one additional substream. Given a choice, for each substream, the node that is the least loaded over all servers that have this substream is chosen. If a serving peer disconnects during the service, the client peer will look for a replacement serving peer that has the same substream and sufficient uplink bandwidth.

The experiments are applied onto four different architectures: In all configurations a 100-node transit-stub network topology generated with GT-ITM [63] is used as the backbone. 9 leaf nodes are attached (1 ultrapeer and 8 leaf peers) to each stub node. The selection criterias are based on the studies carried by [33]. During the second phase of the simulations, two CDN nodes are also connected to the core of the transit-stub network.

In all simulations, one of every nine node is selected as Ultrapeers, and the remaining nodes are configured to be Leaf nodes. The Leaf nodes are divided into two different categories: freeloaders or volunteer sharers. The freeloaders do not let the other peers use the resources they have, on the other hand, volunteer sharers let the rest to use the content they already have. In the architectural design all leaf nodes can have at most three virtual connections to the ultrapeers or other leaf peers.

5.3.3 Simulation Results

The simulations are performed to see the behavior of different architectures of P2P systems in serverless video sreaming is simulated. Additionally, the CDN contribution and HFTS is also employed to get know the contribution needed for a high quality video streaming by using hybrid P2P systems.

P2P System Performance Measurements

P2P System Performance experiments are applied in two phases. In the first phase, the behavior of different architectures of P2P systems in serverless video streaming is simulated. In the second phase, the CDN contribution and HFTS is also employed to enable high quality video streaming by using hybrid P2P systems.

Table 5.1: Basic summary for the simulation phase I. The number of queries initiated, number of useful hits, average hit time and success ratio for different architectures are summarized.

Duration	15 minutes			
# of peers	900			
# of objects	100			
Req. arrival rate	Uniform, 1.1 on average			
Type of Arch.	Rand.	Time	Interest	Two-way
# of queries	1892	2967	4012	5229
# of useful hits	1534	2401	3437	4458
% success	81.1	80.9	85.7	85.3
avg response time (in seconds)	2.0	0.67	1.35	1.07

The random Gnutella overlay network with leaf/ultrapeer structure is established to be used as a reference point for the remaining parts of the simulations. The first phase of the simulation focuses on the implementation of current state of the art clustering mechanisms based on the geographical location-time based dispersion and interest based dispersion techniques. The final part of the first phase aims to show the effect of hybrid clustering on the performance of the system. The first phase of the simulations tries to evaluate the performance differences of different architectures in the scope of P2P metrics. The parameters used for the simulations and the summary of the results obtained are shown in Table 5.1.

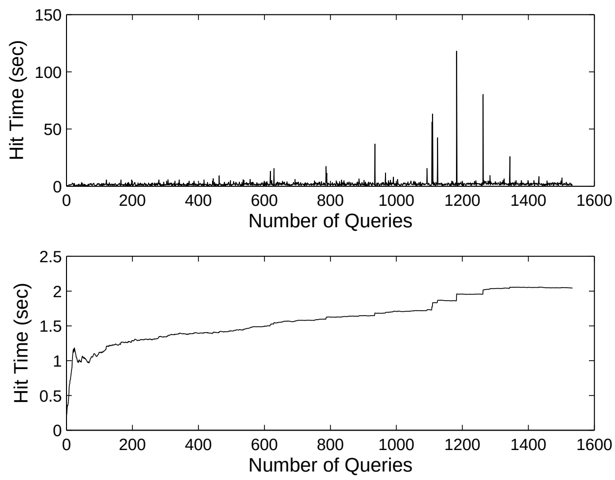


Figure 5.4: Hit time and average hit times for random gnutella architecture

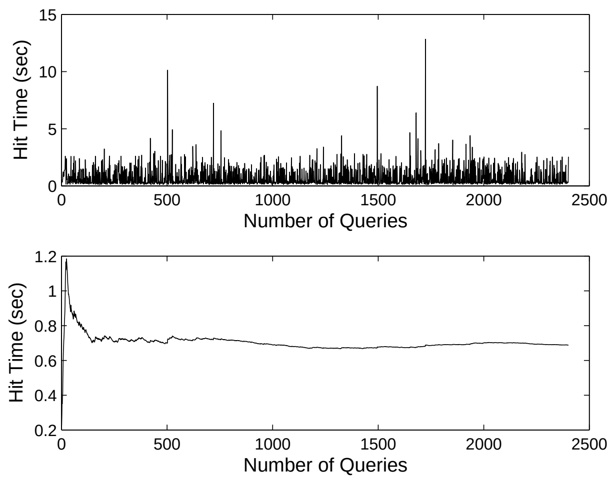


Figure 5.5: Hit time and average hit times for geographical location-time based clustered architecture

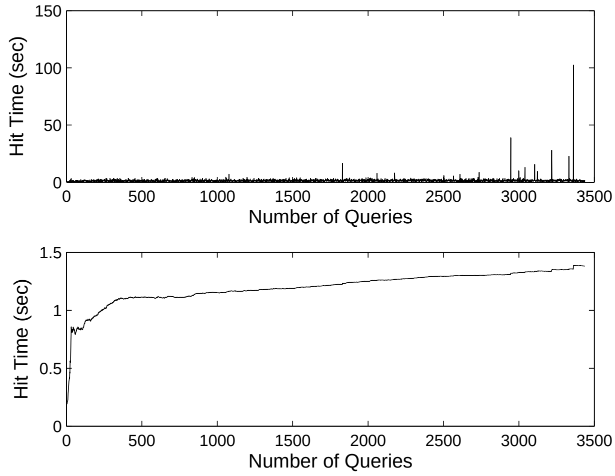


Figure 5.6: Hit time and average hit times for interest based clustered architecture

In the first phase of the experiments, the hit ratios for the queries are measured and the ratio for successfully received (but not necessariliy less then a threshold time) video frames over all the frames are calculated for each simulation. Table 5.1 shows the number of queries, number of responses to the queries and the number of useful responses (unique) to the queries in time. As it can be seen from the table, the hit ratio for the random gnutella architecture and geographical location-time based clustered architecture is almost same. The hit ratio for such system is affected by several factors such as the number of freeloaders and non freeloaders (volunteer sharers) which directly affects the hit ratio. On the other hand, the hit ratios for interest based clustering and the newly offered two-way/hybrid clustering are better than random and geographical location-time based clustered architectures.

Figures 5.4 through 5.7 give the hit times for all queries with the average hit time throughout the first phase of the simulation. The difference between hit times comes from the round trip time and the effective connection capacity between the connected peers. It is also related with the average search depth to locate a peer. It is very small for the geographical location-time based clustering since the mechanism utilizes the network neighborhood of the peers to cluster them. The responses and the packets travel fastest in this architecture since the bandwidth capacity between two peers in this architecture is bigger than the other architectures, and the delay is the smallest compared with the other architectures. Although the peers are spread almost similar in random gnutella and interest based clustering architectures, the interest based clustering architecture performs better than the random gnutella architecture since the chance to locate the searched content in earlier steps is bigger than the one in the random gnutella architecture. In other words, the average search depth to locate a content in interest based clustering in less than the one in the random gnutella architecture. It means that you can easily locate most of the required content within our network neighborhood. The two-way/hybrid clustering performs between two other clustering architectures, since it has the strength of faster locating and delivery of geographical location-time based clustering, and the power of higher success ratio to locate in the interest based architecture which is slower than the geographical location-time based clustering. Thus the average performance of the two-way/hybrid clustering sits between two other clustering architectures. Although, the hit ratios for both random gnutella and geographical location-time based clustered architectures are almost identical, within the same amount of time %50 more queries in the time based clustered architecture is initiated. The major reason for the above result is having the average hit time very small (almost one third) compared with the random gnutella overlay as can be seen from Figures 5.4 and 5.5. Although, the average hit time for two-way/hybrid architecture is in between the time and interest based clustering architectures, the success ratio for the queries is not less than the interest based clustering. Therefore, you can achieve at least the same amount of success within shorter time with the newly offered model.

The hit time distributions for the first phase of the simulation is summarized in Figures 5.8 through 5.11 for different architectures. For the random gnutella architecture, the samples become dense around 2 seconds, however, half of the samples are received after 1.6 seconds. While the hit time for the clustered architectures and newly offered clustering architecture is very small, the number of extraordinary hit times are very limited compared with the random gnutella

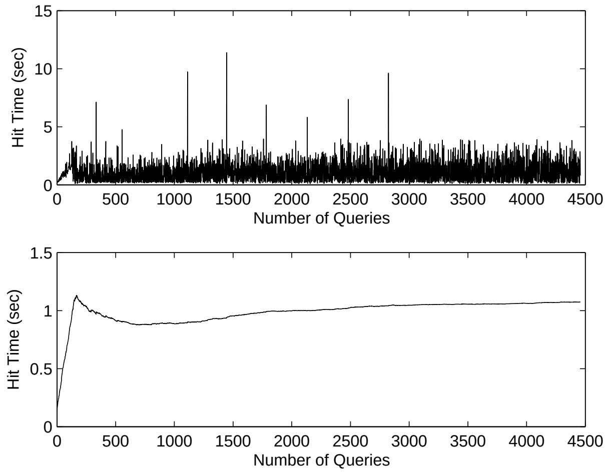


Figure 5.7: Hit time and average hit times for two-way/hybrid clustered architecture

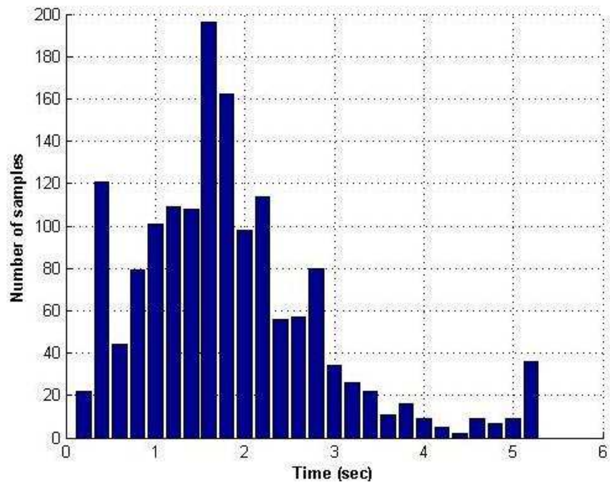


Figure 5.8: The histogram of sample delay distribution for random gnutella architecture

architecture. More than 50% of the hits took less than 0.4 seconds in time based clustering. The interest based clustered architecture has better sample distribution in delivery time than random gnutella architecture; however, it is

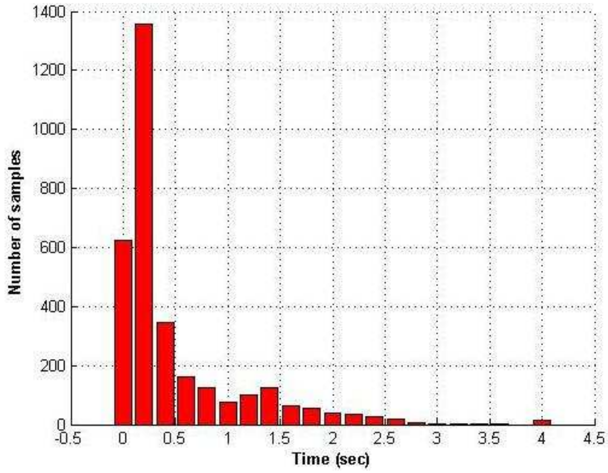


Figure 5.9: The histogram of sample delay distribution for geographical location-time based clustered architecture

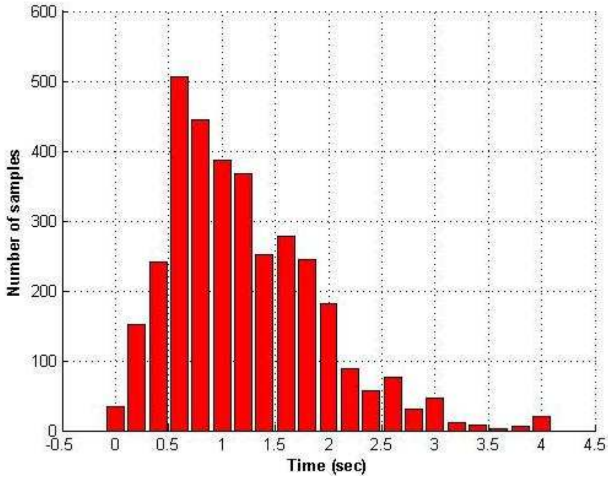


Figure 5.10: The histogram of sample delay distribution for interest based clustered architecture

spread over time. In two-way/hybrid architecture, most of the samples are received within the first second which gives a feasible architecture for video streaming or real time multimedia implementations over P2P systems. As it can be seen from the figures the strength of the geographical location-time based

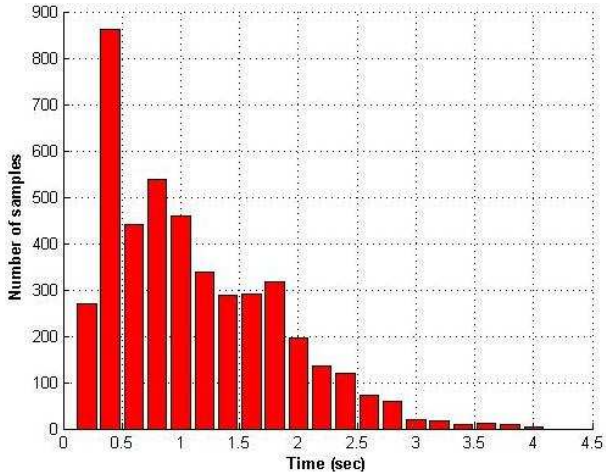


Figure 5.11: The histogram of sample delay distribution for two-way hybrid clustered architecture

clustering comes from the smaller round trip time and the better data exchange capacity in this architecture. The samples in the interest based architecture spread over time, since the locations of the peers in the same cluster would be far from each other according to the time metrics. The two-way/hybrid clustering benefits from the geographical location-time based index first, however if it cannot be located in that index, it utilizes its interest based index to enhance the success ratio. On contrary, this enhancements adds some samples with higher delivery times which increases the average time. Since the time to locate the content in interest based cluster index would take longer than the average time in geographical location-time based clustered architecture.

As it can be seen from the figures 5.8 through 5.11, the random gnutella architecture is not capable enough to be used as a framework in the video streaming applications. From now on, we will use the clustered architectures in the experiments. In this part of the experiments, a video file coded in 30 fps with the duration of 90 seconds is played out in average bandwidth of 1 Mbps. The performance of different architectures are evaluated. Table 5.2 gives a summary of the results obtained from this experiment.

The successfully received frame rate in the Figures 5.12 through 5.14 are calculated as follows: For different buffer sizes, all the received frames are counted, and the ones which does not conform according to the threshold i.e. the reception time is bigger than the threshold is counted as unsuccessful. The threshold value is accepted as the buffer size in duration i.e. 1 second, 2 seconds, and 3 seconds respectively. In every frame, the cumulative number of successfully received frames (# of total frames - # of unsuccessful frames)are divided by the total number of received frames through the time.

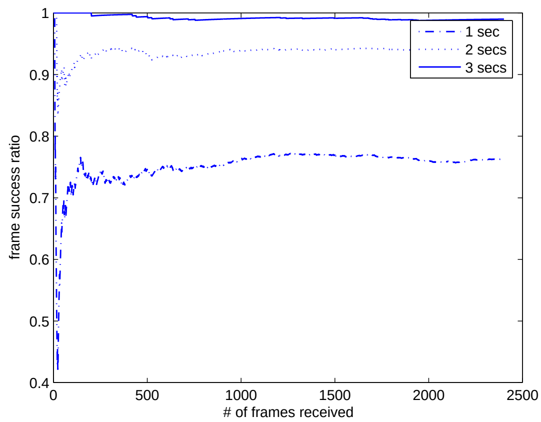


Figure 5.12: The frame success ratio for geographical location-time based clustered architecture with the buffer size of 1,2 and 3 seconds

The ratio for successfully received video frames over all the frames for the first phase of the simulation is summarized in Figures 5.12 through 5.14 for different architectures. Since the geographical location-time based architecture is organized to try to receive the frames from the closest peers, the ratio for successfully received video frames over all the frames for different sized playback buffers performs the best among all different architectures. The reason is, if a frame can be located then it will be received in the shortest time compared with the other architectures.

Although the interest based clustered architecture has better successful frame reception rate than geographical location-time architecture; it has some problems in the ratio for timely received video frames over all the frames since some of the peers are far from the network of the receiver which yields longer delivery

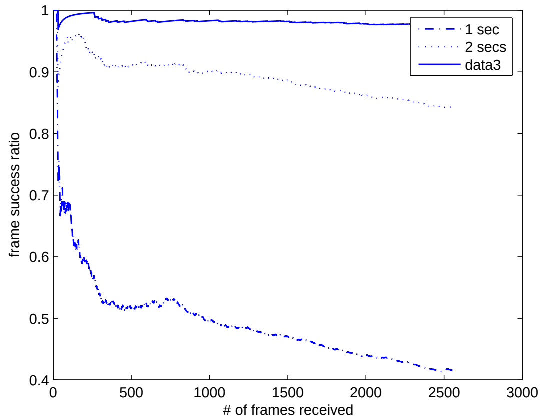


Figure 5.13: The frame success ratio for interest based clustered architecture with the buffer size of 1,2 and 3 seconds

times for the frames since the total time spent between request and reception of the frame is longer than the tolerated amount of time.

The ratio for successfully received video frames over all the frames with a playback buffer of 1 second is about 40% in the long run, which means the successfully received video frames are not suitable to the video streaming applications with this buffer size.

While the two-way/hybrid architecture benefits from both geographical location-time and interest based clustering architectures, it shows the characteristics of both architectures. Although the ratio for successfully received video frames over all the frames for the two-way/hybrid clustering is better than the interest based clustering, it gives worse results than geographical location-time based clustering architecture since the SNR values are calculated for successfully received frames and the reception time for a frame in geographical location-time based clustering architecture is the smallest among the others. However, the success rate in the two-way/hybrid clustering is about 98%, which is the best when compared with the geographical location-time based clustering architecture and interest based clustering architecture.

In the second phase of the simulations the behavior of the designed hybrid CDN-P2P system is simulated and what will happen when a service to a node

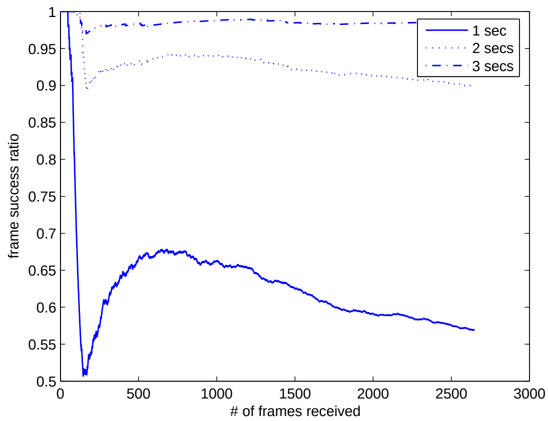


Figure 5.14: The frame success ratio for two-way/hybrid clustered architecture with the buffer size of 1,2 and 3 seconds

fails is modeled. The simulations are based on investigating the effect of using CDNs in the offered architecture with the same video file. When a peer experiences feeding problems with the contributing peers during the streaming session, a query to the CDN system is initiated.

The unsuccessful queries are forwarded to the CDN peers in order to keep the service up. The decision for the forwarding is taken after waiting more than the average hit time plus some safety margin as used in TCP's Exponential Weighted Moving Average System [56]. The method uses the past samples for the packet Round Trip Times (RTT) and estimates the new timeout value by calculating the estimated sample RTT and adding some safety margin which is calculated from the deviations of the RTTs.

According to the CDN experiments, the time required for a successful response to a query varies between 0.1 to 0.3 seconds. The playback buffer of the system is configured as 3 seconds which is suitable for one way streaming applications. The average time required is almost 0.18 seconds. It gives an idea when to forward the unsuccessful queries to the CDN peers. In order to guarantee the required service level to the peers obtaining services from the system, the queries should be forwarded almost 2.7 seconds after the initiation of the query which is the waiting time before backup is used. It will provide

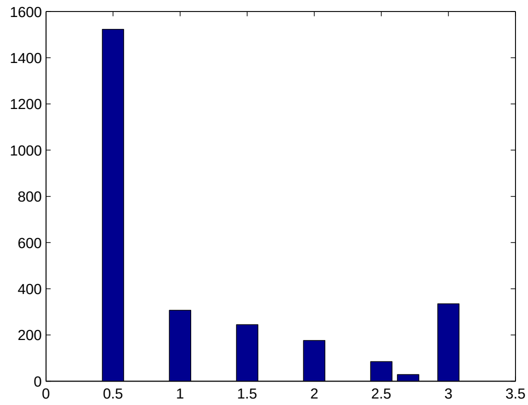


Figure 5.15: The histogram of sample delay distribution for geographical location-time based clustered architecture with CDN forwarding enabled

enough time to locate new peers to get services while there is no loss in quality and service during that search. After activating the CDN contribution support whenever needed, the hit time for any request becomes no more than 3 seconds which is enough for the system to continue with promised streaming quality. This method removes all the spikes when the CDN subsystem is always on.

Although the fault detection time can be calculated adaptively as indicated in the offered model, using the limits of the playback buffer is preferred with having the final chance to get the content from CDN. This method reduces the use of the network resources even further. It also reduces the effect of finding a failure and backup assignment during streaming, since we only wait for a deadline for each packet and if the packet has not been received, the preassigned CDNs are used to get the segment. The cost of running the algorithm is explained in part 5.3.1.

Figures 5.15 through 5.17 show the distribution of sample delays in time for CDN enabled structure. As it can be seen from the figures, the hit time for any query is less than 3 seconds.

Table 5.2 summarizes the effect of CDN contribution with different P2P structures. Although the system guarantees the fault tolerance, it also produces some overhead in the network and CDN peers.

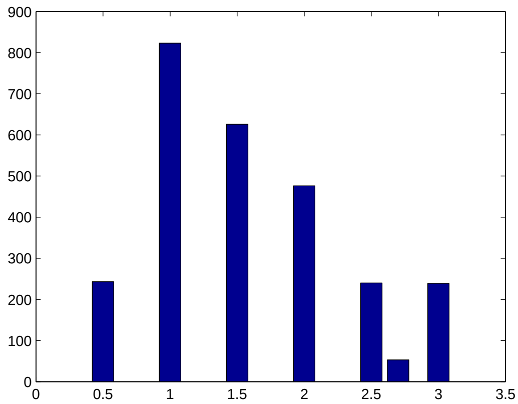


Figure 5.16: The histogram of sample delay distribution for interest based clustered architecture with CDN forwarding enabled

Table 5.2: Basic summary for the simulation phase II, indicating the number of requests handled by CDN servers and its percentage to all requests

Type of Arch.	Time	Interest	Two-Way/Hybrid
total number of frames	2700	2700	2700
number of successful frames	2401	2571	2649
number of unsuccessful frames	299	129	51
CDN/P2P duplicates	36	110	85
Duplicate ratio%	1,3	3,7	3,1

In the time based clustered architecture, 12,4% of all frames are received from the CDN servers. 11% of this comes from the unreceived frames, and the remaining part is coming from delivery after 2.7 seconds. Therefore, almost 1,4% of successful hits are also discarded because of late delivery.

For the interest based clustering architecture, 8,9% of all frames are received from the CDN servers. 5,2% of this comes from the unreceived frames, and the remaining part is coming from delivery after 2.7 seconds. Therefore, almost 3,7% of successful hits are also discarded because of late delivery.

The hybrid/two-way clustering architecture only expects 5% of all frames

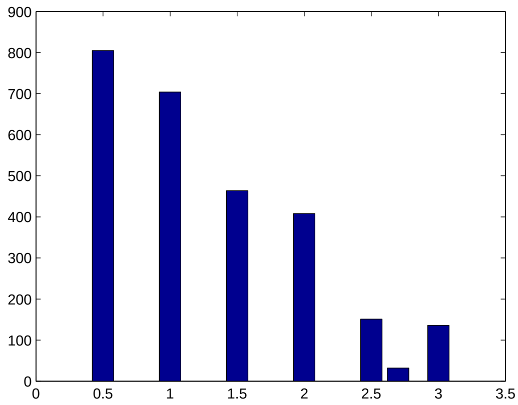


Figure 5.17: The histogram of sample delay distribution for two-way hybrid clustered architecture with CDN forwarding enabled

are received from the CDN servers. 1,9% of this comes from the unsuccessful queries, and the remaining part is coming from delivery after 2.7 seconds. Therefore, almost 3,1% of successful hits are also discarded because of late delivery. Hybrid/Two-way clustering architecture outperforms the nearest architecture by almost 160%.

Thus, the two-way/hybrid clustering architecture utilizes the system resources in the best manner by wasting the useful system resources 6 times less than the geographical location-time based clustered architecture.

5.4 Conclusions of the Chapter

In this chapter, we proposed a fault tolerant hybrid P2P-CDN video streaming architecture. A Two-way/Hybrid clustering method is also introduced in order to increase the search efficiency of the hybrid system.

The capacity of the different clustering architectures are compared in chapter 4. The results show that, the bigger combinatorial capacity is, the better is the utilization of the same communication medium. The architecture having greater channel capacity gives better average hit time and higher number of messages exchanged in the P2P system. The decreasing combinatorial capacity results in longer search results and hit time, and lower number of queries send and hits in the system. The experimental results show that the two-way/hybrid clustering offers the highest channel capacity thus the highest hit ratio and feasible hit time.

The HFTS guarantees to provide total quality stream to a consuming node either only from the P2P overlay or from a hybrid solution consisting of CDN and P2P parts. The offered Two-Way/Hyrid clustering technique outperforms the techniques proposed in the literature by bringing efficient data retrieval with less overhead traffic and fast resource locating and efficient resource usage.

CHAPTER 6

APPLICATIONS OF P2P SYSTEMS

The first part of this chapter aims to show that how the proposed Hybrid Fault Tolerant Video Streaming System can be adapted into application development for educational domain. It firstly gives brief information on learning styles and the suitable learning styles for distance education. It then employ the Three-way/hybrid clustering which uses the learning models-styles offered by Kolb as the third dimension in the clustering phase to optimize the applications for better performance. The second part of the chapter concentrates on extending the modeling of P2P systems with different clustering approaches. It gives an idea on the reliability requirements of different clustering architectures, and the effect of the organization on the reliability requirements of a P2P system. It also gives a mechanism to evaluate the different architectures analytically.

6.1 Learning Styles

Depending on the difference in human characteristics humans/students may exhibit different behaviors in learning mechanisms. Every student has different levels of motivation, different attitudes about teaching an learning, and different responses to specific classroom environments and instructional practices [24]. The early studies on learning is based on the categorization of the learning styles of the students. If the teachers are aware of the differences in the learning styles of the students and use different mechanisms in their teaching methodology, it could give better results in the attitudes of the students toward learning.

There are several studies to model the learning styles of the students, and many different learning models proposed. Felder-Brent [24] states that the best known learning model is the one offered by Myers-Briggs Type Indicator

(MBTI). They argue that while MBTI assesses personality types, but MBTI profiles are known to have strong learning style implications. The models offered by Kolb, and Felder-Silverman are the other popular learning models especially used in engineering education. The learning models use inventories to put the students into a learning style category. [57] states that Kolb Learning Style [46] Inventory (KLSI) is one of the most influential and widely distributed instruments used to measure individual learning preference which is based on the Kolb's experimental learning theory. The learning style is defined in two dimensional scale based on persons perception and processing of information. In the scale prepared by Kolb classified information perception as concrete experience or abstract conceptualization, and information processing is classified as active experimentation or reflective observation [57]. [46] summarizes the learning model offered by Kolb in 1979 includes four learning styles [3] in its inventory:

- *Converger* who can be classified as someone who wants to solve a problem and who relies heavily upon hypothetical-deductive reasoning to focus on specific problems [46].
- *Diverger* who can be classified as someone who solves problems by viewing situations from many perspectives and who relies heavily upon brainstorming and generation of ideas [46].
- *Assimilator* who can be classified as someone who solves problems by inductive reasoning and ability to create theoretical models [46].
- *Accommodator* who can be classified as someone who solves problems by carrying out plans and experiments...and adapting to specific immediate circumstances [46].

In addition to the learning styles, Kolb also offers learning cycle preferences for the students. Kolb defined four learning cycles:

- *Concrete Experience* where learning from feelings [12] or reactions to experience influence your learning.

- *Reflective Observation* where learning from watching and listening [45] influence your learning.

- *Active Conceptualization* where learning from thinking [45] or analyzing problems in a systematic method influence your learning.

- *Active Experimentation* where learning by doing [45] or results driven influence your learning.

The learning cycles and styles have strong relations between them. For example, a converger favors a learning cycle of Abstract Conceptualization and Active Experimentation, which fits since these two learning cycles are characterized by learning by doing and thinking [3]. According to Kolb [46], the learning style of a student can change from a style to another over time. As noted before, the model offered by Felder-Silverman [25] is also very popular in engineering education. They categorize the learning styles of students and claim that if there exists a mismatch between learning styles of the students in a class and the teaching style of the professor, the students and the system may probably fail to succeed. They point out the solution of such problems to balance the instructional methods of the professors with the preferences of the students. The learning models are developed to solve the communication problems in the same location. On the other hand, the evolution in communication helped us to develop applications providing distance learning over Internet. The e-learning projects willing to succeed should consider the learning styles of the potentional learners. The learning models shall be modified to comply all the requirements in e-learning environments. [13] points out two major methods to find out the learning model of a person via hypermedia that are: collaborative and automatic user modelling. The collaboravie model uses and relies on answers of the users to the questionnaire for assessing preferences. On the other hand, the automatic user modelling uses the behavior and actions of the users adaptively. [26] developed an Index of Learning Style (ILS) self-scoring questionnaire to be used for adaptive multimedia systems in order to identify the learning styles of the users based on the Felder-Silverman model. [14, 29, 47, 83] agree that Felder-Silverman model seems to be the most suitable one to be used in computer-based educational systems. Additionally, [29,83] offer an automatic student modelling approach which starts with a questionnaire, then, identifies the students' learning styles based on their actual behaviour and actions in the online course. They developed their approach especially to be used for learning management systems (LMS) for e-learning. They also categorize the learners behaviors into four dimensions based on the Felder-Silverman model. The categories are as follows: active/reflective way of processing information, sensing the facts or intiutively learn by using abstract materials, using visual objects or verbal texts and learning step by step (sequential) or learning in large bursts (global).

6.2 Clustering Based on Learning Styles

The Kolb Model and Felder-Silverman Model are the most popular Learning models adapted into engineering education. They are also successfully applied into e-learning applications. Although, the proposed Three-way/hybrid Fault Tolerant Streaming System is based on learning model offered by Kolb, the same architecture can also be implemented on Felder-Silverman Model or similar learning models. The study aims to provide a collaborative learning tool based on P2P approach including textual, visual, and interactive learning content. In the proposed system, the Two-way/hybrid clustering technique is extended to have additional clustering level running on top of interest based clustering. The peers (students) are connecting to Internet from any location. In the study, the peers (students) are clustered by considering: (1) the geographical location they are connecting from, (2) the interest they have i.e. the lecture they are willing to study, and (3) the learning style-cycle they are grouped. The proposed system uses HFTS algorithm offered in the previous chapter with replacement of two-way/hybrid clustering to three-way/hybrid clustering architecture.

In the proposed model, there are two levels of interest clusters (and additional geographical location-time based clustering). In the base interest cluster, peers are clustered together according to their interests. In the second level interest cluster, peers are clustered together according to their learning styles-cycles. The new model helps the user to differentiate the materials in the same interest group like Active experimenters, Reflective observers and tries to query the peers having the same learning preferences first in the same interest category.

Figure 6.1 shows the basic components of the proposed system including peers, three-way clustering layout and the traffic flow amongst the system components.

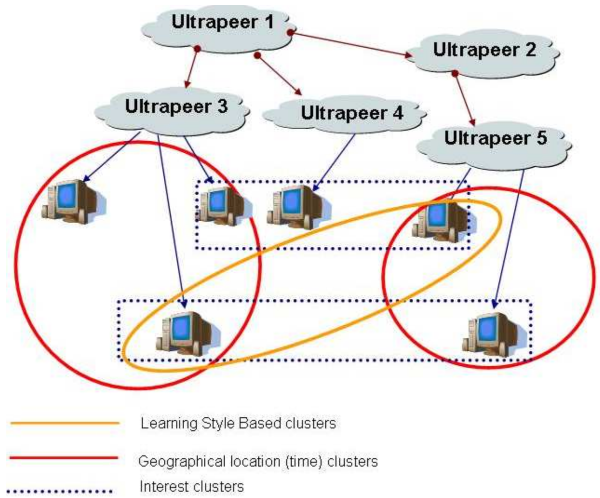


Figure 6.1: The logical organization of a typical Three-way clustered overlay network.

One of the main drawback of the offered system would be the overhead of dispersing the peers by using three-way clustering. The overhead caused by including three-way clustering is not more than the cost caused by two-way clustering during the dispersion phase and control exchange during the execution. The additional level of interest cluster based on learning model keeps an additional label to differentiate the learning model of each peers in the same interest cluster. The additional heart beat messages are negligible when compared with the base heart beat messages exchanged to form two-way/hybrid clusters because of the limited amount of additional overhead generated.

6.2.1 Experimental Results

The simulation settings are almost the same with the experiments conducted for HFTS studies except for the additional interest level clusters consisting of four different learning preferences: namely Active experimenters, Reflective observers, Concrete experiencers and Active conceptualizers. The learning preferences of the peers are initially set by the system then they are adaptively adjusted by considering the behaviors of the peers during the simulation. As noted by Kolb [46], learning preference of a student may change from time to time, therefore, we allowed the peers adaptively adjust their preferences.

The simulations are performed to see the behavior of different architectures of P2P systems in serverless video streaming. Additionally, the CDN contribution and HFTS is also employed to get the contribution needed for a high quality video streaming by using hybrid P2P systems.

Three-way/hybrid clustering System Performance experiments are applied in two phases. In the first phase, the behavior of different architectures of P2P systems in serverless video streaming is simulated. In the second phase, the CDN contribution and HFTS is also employed to get the contribution needed for a high quality video streaming by using hybrid P2P systems.

The Two-way/hybrid clustering approach is established to be used as a reference point to get the effect of additional level of interest cluster based on the learning modelś effects on the systems performance. The simulations try to evaluate the performance differences of different architectures in the scope of P2P metrics. The parameters used for the simulations and the summary of the results obtained are shown in Table 6.1.

Table 6.1: Basic summary for the simulation for Three-way/hybrid clustering in e-learning applications

Duration	15 minutes	
# of peers	900	
# of objects	100	
Req. arrival rate	Uniform, 1.1 on average	
Type of Arch.	Three-way	Two-way
# of queries	5027	5229
# of useful hits	4424	4458
% success	87.9	85.3
avg response time (in seconds)	1.088	1.07

In the first phase of the experiments, the hit ratios for the queries are measured and the ratio for successfully received video frames over all the frames are calculated for each simulation. Table 6.1 shows the number of queries, number of responses to the queries and the number of useful responses (unique) to the queries in time. According to the experiments, the newly designed algorithm handles the same amount of successful operations with less queries performed.

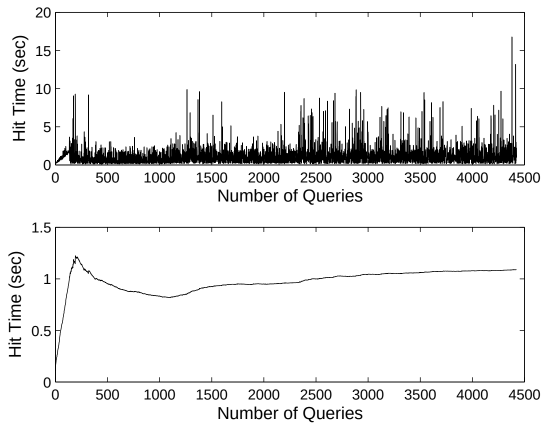


Figure 6.2: Hit time and average hit times for three-way/hybrid clustered architecture

As it can be seen from the table, the hit ratio for three-way clustering is better than the two-way/hybrid clustering architecture. Additionally, Figure 6.2 gives the hit times for all queries with the average hit time throughout the first phase of the simulation. Although, three-way clustering performs best in success ratio in all the clustering architectures, the average hit time is slightly worse than two-way/hybrid clustering.

The hit time distributions for the first phase of the simulation is summarized in Figure 6.3 three-way/hybrid clustering architecture. Most of the samples are received within the first second as it were in the two-way/hybrid architecture, which gives a feasible platform for video streaming or real time multimedia implementations over P2P systems.

As it was in the two-way/hybrid clustering, three-way hybrid clustering benefits from the geographical location-time based index first, however if a requested

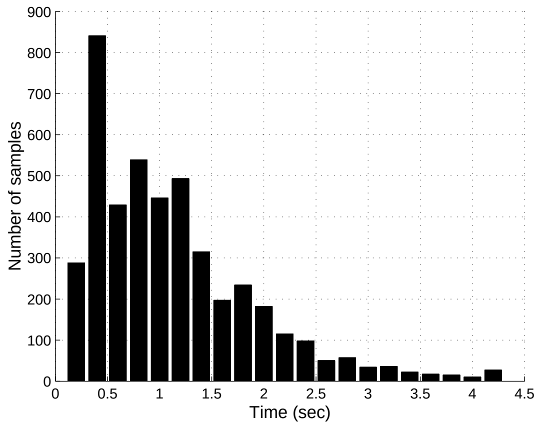


Figure 6.3: The histogram of sample delay distribution for three-way /hybrid clustered architecture

content cannot be located using that index, the proposed architecture utilizes its interest based index to enhance the success ratio. Additionally, it also employs the second level index clusters to obtain the samples faster. On the contrary, these enhancements adds some samples with higher delivery times which increases the average delivery time. The samples received in more than one second are mostly received from the nonrelated peers or the peers from the same interest clusters. The higher success ratio in this architecture is the result of having an additional level of interest clusters which also results in longer average hit time compared with the two-way/hybrid clustering.

In the second part of the experiments, a video file coded in 30 fps with the duration of 90 seconds is played out as we did for the other clustering architectures. The performance evaluation of three-way/hybrid clustering architecture obtained from this experiment is summarized in Table 6.2 with the comparison of the results obtained from two-way/hybrid clustering architecture.

The successfully received frame rate in the Figure 6.4 is calculated in the same manner as it is done in the previous chapter: for different buffer sizes, all the received frames are counted, and the ones which do not conform according to the threshold i.e. the reception time is bigger than the threshold, is counted as unsuccessful. The threshold value is accepted as the buffer size in duration i.e.

1 second, 2 seconds, and 3 seconds respectively. In every frame, the cumulative number of successfully received frames (# of total frames - # of unsuccessful frames) are divided by the total number of received frames through the time.

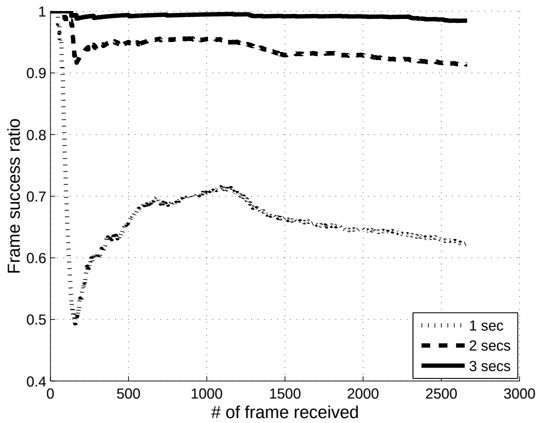


Figure 6.4: The frame success ratio for three-way/hybrid clustered architecture with buffer size of 1,2 and 3 seconds

While the three-way/hybrid architecture benefits from two-way/hybrid clustering architecture with an additional level of interest cluster, it shows similar characteristics with two-way/hybrid clustering architecture. Although, still the geographical location-time based clustering architecture performs best on the SNR values calculated for successfully received frames and the reception time for small buffer lengths, the success rate in the three-way/hybrid clustering architecture for the buffer size of 3 seconds is about 99%, which is the best when compared with all the other clustering architectures.

In the second phase of the simulations, the behavior of the designed hybrid CDN-P2P system is simulated with three-way/hybrid clustering architecture and what will happen when a service to a node fails is modeled. The simulations are based on investigating the effect of using CDNs in the offered architecture with the same video file. When a peer experiences feeding problems with the contributing peers during the streaming session, a query to the CDN system is initiated as it is done for all the clustering architectures in the previous chapter. The cost of running the HFTS algorithm with three-way/hybrid clustering algorithm is almost the same as the cost of implementing HFTS algorithm with two-way/hybrid clustering algorithm.

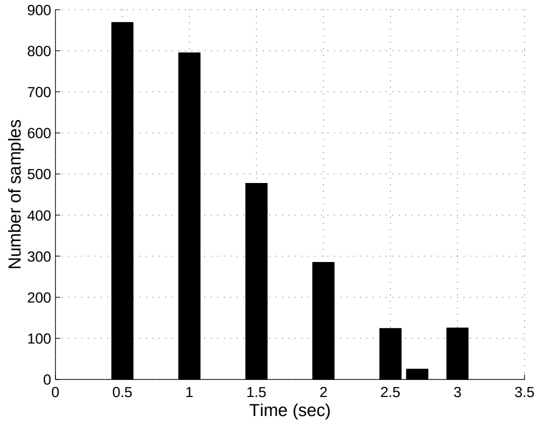


Figure 6.5: The histogram of sample delay distribution for three-way/hybrid clustered architecture with CDN forwarding enabled

Figure 6.5 shows the distribution of sample delays in time for CDN enabled structure. As it can be seen from the figure, the hit time for any query is less than 3 seconds. Since, it is guaranteed by the HFTS algorithm.

Table 6.2: The number of requests handled by CDN servers and its percentage to all the requests for two-way and three-way/hybrid clustering architectures

Type of Arch.	Three-way/Hybrid	Two-Way/Hybrid
total number of frames	2700	2700
number of successful frames	2665	2649
number of unsuccessful frames	35	51
CDN/P2P duplicates	90	85
Duplicate ratio%	3,3	3,1

Table 6.2 summarizes the effect of CDN contribution with two-way/hybrid and three-way/hybrid clustering architectures. Although the system guarantees the fault tolerance, it also produces some overhead in the network and CDN peers.

While the two-way/hybrid clustering architecture only expects that 5% of all frames are received from the CDN servers, it is 4.6% for three-way/hybrid clustering architecture. In the three-way/hybrid clustering architecture, 1.3% of this comes from the unsuccessful queries, and the remaining part is coming from delivery after 2.7 seconds. Almost 3.3% of successful hits are also discarded because of late delivery.

Therefore, the proposed method considering the *Learning Style* of the peers (three-way/hybrid clustering architecture) utilizes the system resources in the best manner by performing 10% better than the two-way/hybrid clustering architecture when used as a platform under QoS aware video streaming systems.

6.3 Modeling Different Clustering Architecture

In this part, the model for measuring the system reliability with k back-up connections is applied to different clustering architectures. In the model, rather than considering the number of redundant copies in the network, the number of available peers in the system with certain success probabilities has been defined.

In [88], to overcome the degradation while locating a new peer for the replacement of a failed one, in order to make sure not to lead to unsatisfactory quality, we offered sub-stream encoding mechanisms with error control such as FEC or MD-FEC so that a receiver tries to maintain $M = L + K$ sub-streams. The goal is to model and quantify the impact of the parameter K on system resilience to churn. In the model, the parameter $\bar{\lambda}$ denotes the rate at which the client can connect to one of the available peers and is used for modeling the query time to find a peer that has the media and to setup the associated connection.

6.3.1 Numerical Evaluation

In order to easily evaluate and compare the behavior of different clustering algorithms, the success probabilities with respect to time and the MTTF values are evaluated using fixed parameter sets. The average success time for different clustering algorithms are used for the $\bar{\lambda}$ values to calculate the success probabilities and the MTTF values for different architectures. During the evaluations, we use $L = 4$ seeding peers, and 100 as the number of peers in the system. The number of peers in the system with 20% success probability for any peer (it is used throughout all simulations) gives different N values for each architecture. The number of extra connections is set to 1 , with $1/\bar{\lambda} = 2, 0.67, 1.35, 1.07$ and 1.088 seconds respectively reflecting the random gnutella, geographical location-time based, interest based, two-way/hybrid [64] and three-way/hybrid clustering architectures.

The cases corresponding to $1/\lambda = 1/\mu = 4, 16$, and 64 minutes are studied and the resulting success probabilities $S(t)$ are depicted in Figures 6.6, 6.7 and 6.8 as a function of the video length t, and the MTTF values are presented in Table 6.3.

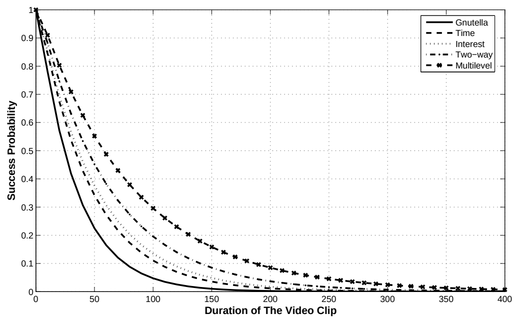


Figure 6.6: Success probability for a video clip with varying length when $1/\lambda = 1/\mu = 4$ minutes

We have two main observations:

- When the dynamicity decreases, ($1/\lambda = 1/\mu = 64$ minutes) the effect of $\bar{\lambda}$ decreases. Since the probability of failure with the current connection decreases. The migration from dynamic systems to more stable systems gives good results in the reliability of the system in terms of $S(t)$ and MTTF.

- When dynamicity decreases, the effect of having more redundant copies are observed more clearly while dynamicity increases the effect of $\bar{\lambda}$ is more

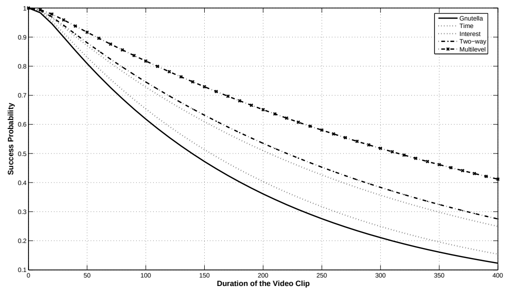


Figure 6.7: Success probability for a video clip with varying length when $1/\lambda = 1/\mu = 16$ minutes

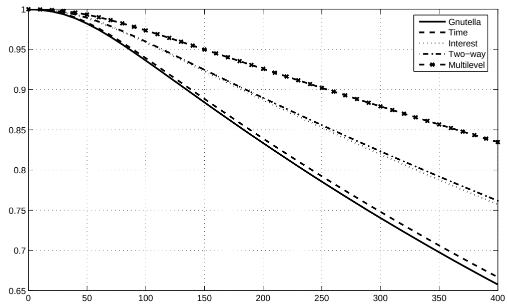


Figure 6.8: Success probability for a video clip with varying length when $1/\lambda = 1/\mu = 64$ minutes

meaningfull compared with the effect of the number of redundant copies.

6.4 Conclusions of the Chapter

In this chapter, a research is conducted on the learning styles of the students. The studies indicate that, the preferences of the learner shall be considered in order to obtain better results. The HFTS (including two-way/hybrid clustering algorithm) offered in chapter 5 is improved to contain the learning models of the people as the third dimension in the overlay network organization. The new clustering algorithm used in HFTS is called three-way/hybrid clustering architecture which gives better results in performance of the system.

Table 6.3: MTTF Values for Model B in Minutes for different clustering architectures with different characteristics (Random gnutella, interest and time based clustering, two-way/hybrid and three-way/hybrid clustering architectures are evaluated)

$1/\lambda, 1/\mu$ minutes	$Random$	$Time$	$Interest$	$Two-way$	$Three-way$
$1/\lambda = 1/\mu = 4$ min.	34.1	46.69	50.99	62.26	82.46
$1/\lambda = 1/\mu = 16$ min.	196.28	218.95	291.11	312.36	449.26
$1/\lambda = 1/\mu = 64$ min.	888.84	916.17	1310.45	1336.38	1980.73

In the second part of the chapter, we tried to compare the performance of different clustering algorithms by modeling them with the reliability modeling offered in Chapter 3. The numerical evaluation results obtained from the reliability modeling overlaps with the results obtained from conventional P2P evaluation metrics.

CHAPTER 7

CONCLUSIONS

The previous chapters explained the research problems considered in this thesis and our approaches in addressing them.

The chapter is organized as follows. Section 7.1 discusses the conclusions drawn from the research conducted in this thesis. The section basically starts with the benefits gained from the new metric. Then, the methods discussed how to bring fault tolerance to P2P video streaming evaluated. The effect of clustering the substrate and the comparison of the current clustering methods and the offered two-way/hybrid clustering are compared. Next, the cost and benefits from applying th total solution for the P2P video streaming with Fault Tolerant schemes are evaluated. Finally, the proposed HFTS is applied to learning styles to obtain better results in e-learning environments by using the three-way/hybrid clustering architecture. Section 7.2 sketches possible directions for extending this research for future studies.

7.1 Overview of the Main Contributions

In Chapter 3, the reliability issue in a P2P system especially for video streaming applications is investigated. The behavior of a peer in a P2P environment is modeled by considering the characteristics. As noted erlier, the major problems in P2P architectures is peer dynamicity and churn. The offered methods to overcome the peer dynamicity and churn has been analyzed with detailed simulations. Finally, the studies combines using Redundancy Free MDC and the peer behavior in a P2P environment to have redundant and reliable P2P video streaming.

In Chapter 4, a metric for P2P networks based on Shannon's L-channel capacity calculation idea is proposed. The metric calculates the maximum rate of information (in bits per second) that can be transmitted over P2P network (a.k.a. combinatorial capacity) caused by protocol and overlay-level connectivity. Metric is applied to the current clustering approaches and newly offered clustering method – two-way/hybrid clustering. The obtained capacity results for different architectures show that the metric has better results identifying the capabilities of the overlay network when compared with the results obtained from the current metrics used in P2P world, e.g. hit time, success ratio, and other metrics such as *turnover* and *size-change*.

Chapter 5 is the most important part of this thesis. In the chapter, firstly a new clustering approach – two-way/hybrid clustering architecture which benefits from the advantages of both geographical location-time and interest based clustering is proposed. The new model has been compared with geographical location-time based and interest based clustering approaches. The simulation results show that the offered architecture outperforms all of the currently available P2P architectures. Then, the state of the art solution to P2P video streaming Hybrid Fault Tolerant Streaming (HFTS) is offered. The HFTS system guarantees to provide expected quality of service in streaming to a consuming node either only from the P2P overlay or from a hybrid solution consisting of CDN part and P2P parts. The main idea implementing HFTS is, although there are enough number of peers in any P2P system to have fault tolerance for hundreds of years as shown in reliability model, there are some resource locating problems in the system, because of the nature of the P2P system. Hence, a hybrid solution to guarantee the fault tolerance with low operational costs to manage is offered.

Chapter 6 extends the studies to application area. The newly designed Fault Tolerant, highly scalable platform for P2P video streaming is applied into educational model by inventing a three-way/hybrid clustering architecture to be used in e-Learning platforms. By employing the three-way/hybrid clustering in learning models, the online distance education systems can be optimized for better performance with HFTS algorithm. Finally, the reliability model offered in chapter 3 is applied to all the known clustering algorithms to analyze their performances quantitatively. The effect of organization on the reliability requirements of a P2P system is obtained. The results of the analytical studies show that, the newly offered two-way/hybrid and three-way/hybrid clustering algorithms performs better than the current clustering algorithms.

7.2 Future Studies

There are several topics untouched in this study. First of all, grading of the services taken from the other peers would be implemented. Secondly, security concerns are not covered at all. Finally, revenue sharing mechanisms would be implemented to make additional benefits for better contribution in the system, to force the peers not only consider their rationality but also the reality of the system.

The future studies shall combine the above statements to build fully functional, fault tolerant streaming applications over P2P systems.

REFERENCES

[1] Akamai home page. [online-last visited on 31st of January,2008]. Available at http://www.akamai.com.

[2] Digital island home page. [online-last visited on 31st of January,2008]. Available at http://www.digitalisland.com.

[3] Learning theory: Using kolb's learning style inventory with computer based training. [online-last visited on 31st of January,2008]. Available at http://www.chartula.com/LEARNINGTHEORY.PDF.

[4] YouTube. [online-last visited on 31st of January,2008]. Available at http://www.youtube.com.

[5] Joint scalable video model (jsvm) 4.0 reference encoding algorithm description. Technical Report ISO/IEC JTC 1/SC 29/WG 11, N7556, ISO, October 2005.

[6] Gnutella home page. [online-last visited on 31st of January,2008] on http://www.gnutella.com, 2007.

[7] Kazaa home page. [online-last visited on 31st of January,2008] on http://www.kazaa.com, 2007.

[8] Napster home page. [online-last visited on 31st of January,2008] http://www.napster.com, 2007.

[9] E. Akyol, A. M. Tekalp, and M. R. Civanlar. A flexible multiple description coding framework for adaptive peer-to-peer video streaming. *IEEE Journal of Selected Topics in Signal Processing*, 1(2):231–245, August 2007.

[10] R. Bhagwan, S. Savage, and G. Voelker. Understanding availability. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, Berkeley, CA, USA, February 2003.

[11] B.Krishnamurthy, C.Wills, and Y. Zhang. On the use and performance of content distribution networks. In *ACM SIGCOMM Internet Measurement Workshop*, San Francisco, USA, November 2001.

[12] R.E. Broyatzis and D.A. Kolb. From learning styles to learning skills: The executive skills profile. *Journal of Managerial Psychology*, 10(5):3–17, 1995.

[13] P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.

[14] C.A. Carver, R.A. Howard, and W.D. Lane. Addressing different learning styles through course hypermedia. *IEEE Transactions on Education*, 42(1):33–38, 1999.

[15] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: High-bandwidth multicast in cooperative environments, 2003.

[16] X. Cheng, C. Dale, and J. Liu. Understanding the characteristics of Internet short video sharing: YouTube as a case study. *ArXiv e-prints*, 707, July 2007.

[17] Y. Chu, A. Ganjam, T. Ng, S. Rao, K. Sripanidkulchai, J. Zhan, and H. Zhang. Early experience with an internet broadcast system based on overlay multicast, 2003.

[18] T. Condie, S. D. Kamvar, and H. G. Molina. Adaptive peer to peer topologies. In *International Conference on P2P Computing*, August 2004.

[19] J. Crowcroft and I. Pratt. Peer to peer: Peering into the future, advanced lectures on networking 2002 tutorials. *Lecture Notes in Computer Science*, 2497, May 2002.

[20] S. S. Kulkarni D. Xu, C. Rosenberg, and H. K. Chai. A cdn-p2p hybrid architecture for cost effective streaming media distribution. *ACM/Springer Multimedia Systems Journal*, 11(4), 2006.

[21] M. Deshpande and M. Venkatasubramanian. The different dimensions of dynamicity. In *Proceedings of the Fourth International Conference on Peer-to-Peer Computing (P2P'04)*, pages 244–251, Zurich, Switzerland, August 2004.

[22] C. Duma, N. Shahmehri, and G. Caronni. Dynamic trust management for peer-to-peer systems. In *In Proc. of 16th International Workshop on Database and Expert Systems Applications IEEE DEXA05*, 2005.

[23] F.B.Kashani and C.Shahabi. Criticality-based analysis and design of unstructured peer-to-peer networks as "complex systems". In *In the Proceedings of the 3st International Symposium on Cluster Computing and the Grid*, 2003.

[24] R.M. Felder and R.Brent. Understanding student differences. *Journal of Engineering Education*, 94(1):57–72, 2005.

[25] R.M. Felder and L.K. Silverman. Learning and teaching styles in engineering education. *Engineering Education*, 78(7):674–681, 1988.

[26] R.M. Felder and B.A. Soloman. Index of learning styles questionnaire, 2008.

[27] P. Francis, S. Jamin, V. Paxson, L. Zhang, D.F. Gryniewicz, and Y. Jin. An architecture for a global internet host distance estimation service. In *Proceedings of IEEE INFOCOM '99*, March 1999.

[28] Vivek K. Goyal. Multiple description coding: Compression meets the network. *IEEE Signal Processing Magazine*, 18(5):74–93, September 2001.

[29] S. Graf, S.R. Viola, and Kinshuk. Automatic student modelling for detection learning style preferences in learning management systems. In *IADIS International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2007)*, December 2007.

[30] C. Griwodz, M. Bar, and L. C. Wolf. Long-term movie popularity models in video-on-demand systems or the life of an on-demand movie. In *The fifth ACM International Multimedia Conference*, 1997.

[31] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica. The impact of dht routing geometry on resilience and proximity. In *In Proc. of SIGCOMM03*, August 2003.

[32] H. Guo, K. T. Lo, and C. T. Cheng. Overlay networks construction for multilayered live media streaming. In *Eighth International Symposium on Multimedia (ISM'06)*, pages 427–436, San Diego, CA,USA, December 2006.

[33] Q. He, M. Ammar, G. Riley, H. Raj, and R. Fujimoto. Mapping peer behavior to packet-level details: A framework for packet-level simulation of peer-to-peer systems. In *MASCOTS 2003*, Orlando,Florida,USA, October 2003.

[34] M. Hefeeda. *A Framework For Cost-Effective Peer-To-Peer Content Distribution*. PhD thesis, Purdue University, August 2004.

[35] M. Hefeeda and B. Bhargava. On-demand media streaming over the internet. In *Proceedings of the 9th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03)*, 2003.

[36] M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava. Promise: Peer-to-peer media streaming using coollectcast. In *In Proc. of ACM Multimedia 2003*, pages 45–54, November 2003.

[37] M. Hefeeda, A. Habib, D. Xu, and B. Bhargava. Collectcast: A tomography-based network service for peer-to-peer streaming. In *In ACM SIGCOMM'03 Poster Session*, August 2003.

[38] M. Hefeeda, A. Habib, D. Xu, B. Bhargava, and B. Botev. Collectcast: A peer-topeer service for media streaming. Technical Report CSD-TR 03-016, Department of Computer Sciences, Purdue University, 2003.

[39] K. Hildrum and J. Kubiatowicz. Asymptotically efficient approaches to fault-tolerance in peer-to-peer networks. In *Proceeding of 17th International Symposium on Distributed Computing*, pages 321–336, October 2003.

[40] S. M. Hotz. *Routing information organization to support scalable interdomain routing with heterogeneous path requirements*. PhD thesis, University of Southern California, 1994.

[41] The MathWorks Inc. MATLAB Software Ver. 7.2.0 Manual, 2006.

[42] X. Jiang, Y. Dong, D. Xu, and B. Bhargava. Gnustream: a p2p media streaming system prototype. In *IEEE International Conference on Multimedia and Expo*, July 2003.

[43] A. Khandekar, R. McEliece, and E. Rodemich. The discrete noiseless channel revisited. Coding, Communications, and broadcasting, P. Farrel, M. Darnell, and B. Honary, Eds. Baldock, U.K.: Research Studies Series Ltd., 2000.

[44] H. Kim, S. Kang, and H. Y. Yeom. Node selection for a fault-tolerant streaming service on a peer-to-peer network. In *Proceedings of the 4th IEEE ICME'2003*, July 2003.

[45] D.A. Kolb. Lsi learning-style inventory. Technical report, McBer & Company, Training Resources Group, 116 Huntington Avenue, Boston, 1985.

[46] D.A. Kolb, I.M. Rubin, and J.M. McIntyre. *Organizational Psychology, A Book of Readings*. Prentice-Hall, New Jersey, 1979.

[47] J. Kuljis and F. Liu. A comparison of learning style theories on the suitability for elearning. In *Proceedings of the IASTED Conference on Web Technologies, Applications, and Services*, pages 191–197, 2005.

[48] R. Kumar, Y. Liu, and K. Ross. Stochastic fluid theory for p2p streaming systems. In *INFOCOM*, pages 919–927, Anchorage, Alaska, 2007.

[49] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytical Methods in Stochastic Modeling*. ASA-SIAM Series on Statistics and Applied Probability, 2002.

[50] Jack Y.B. Lee and W.T. Leung. Design and analysis of a fault tolerant mechanism for a server-less video-on-demand system. In *ICPADS*, page 489, Taiwan, December 2002.

[51] Jin Li. Peerstreaming: An on-demand peer-to-peer media streaming solution based on a receiver-driven streaming protocol. In *IEEE 7th Workshop on Multimedia Signal Processing*, October 2005.

[52] W. Li. Overview of fine granularity scalability in MPEG-4 video standard. *IEEE Trans. Circuit and System for Video Technology*, 11(3):301–317, 2001.

[53] David Liben-Nowell, Hari Balakrishnan, and David Karger. Analysis of the evolution of peer-to-peer systems. In *PODC '02: Proceedings of the twenty-first annual symposium on Principles of distributed computing*, pages 233–242, New York, NY, USA, 2002. ACM Press.

[54] D. K. Liu and R. H. Hwang. P2broadcast: A hierarchical clustering live video streaming system for p2p networks. In *Computer Communications and Networks, 2003. ICCCN 2003*, pages 115– 120, USA, October 2003.

[55] Zhengye Liu, Yanming Shen, Shivendra Panwar, Keith W. Ross, and Yao Wang. Efficient substream encoding for P2P video, 2006. submitted.

[56] L.L.Peterson and B.S.Davie. *Computer Networks*. Maurgan Kaufman Publishers, USA, 2003.

[57] H. Lu, L. Jia, S.H. Gong, and B. Clark. The relationship of kolb learning styles, online learning behaviors and learning outcomes. *Educational Technology & Society*, 10(4):187–196, 2007.

[58] D. W. Manchala. E-commerce trust metrics and models. *Internet Computing*, 4(2):36–44, 2000.

[59] D. E. Meddour, M. Mushtaq, and T. Ahmed. Open issues in p2p multimedia streaming. In *IEEE ICC 2006 Workshop on Multimedia Communications Workshop (MultiCom)*, Istanbul, Turkey, June 2006.

[60] M. F. Neuts. *Structured stochastic matrices of M/G/1 type and their applications*. Marcel Dekker, New York, 1989.

[61] T.S. Eugene Ng and H. Zhang. Predicting internet network distance with coordinates-based approach. In *INFOCOM 2002 Twentyfirst Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 22, pages 170–179, August 2002.

[62] T. Nguyen and A. Zakhor. Distributed video streaming over internet. In *In Proc. of Multimedia Computing and Networking (MMCN02)*, January 2002.

[63] T. Nguyen and A. Zakhor. Distributed video streaming with forward error correction. In *In Proc. Int'l PacketVideo Workshop (PV'02)*, April 2002.

[64] K. Öztoprak and G. B. Akar. Two-way/hybrid clustering architecture for peer to peer systems. In *Second International Conference on Internet and Web Applications and Services (ICIW'07)*, Morne, Mauritius, May 2007.

[65] K. Öztoprak and G.B. Akar. Two-way/hybrid clustering architecture for peer to peer systems. *submitted to IEICE Transactions on Communications available on http://www.ceng.metu.edu.tr/k̃asim/papers/ieice.pdf*, 2007.

[66] K. Öztoprak and H. Kılıç. Protocol and connectivity based overlay level capacity calculation of p2p networks. In *EEE WC&ACM International Workshop on P2P Computing and Autonomous Agents (P2PAA 2006)*, December 2006.

[67] R. Puri, K. Ramchandran, K. W. Lee, and V. Bharghavan. Forward error correction (FEC) codes based multiple description coding for Internet video streaming and multicast. *Signal Processing: Image Communication*, 16(8):745–762, 2001.

[68] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable contentaddressable network. In *In Proc. of ACM SIGCOMM01*, August 2001.

[69] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *In Proc. of 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware01)*, November 2001.

[70] K. Samant and S. Bhattacharyya. Topology, search, and fault tolerance in unstructured p2p networks. In *37th Hawai International Conference on System Sciences*, 2004.

[71] S. Saroiu, P. K. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. Technical Report #UW-CSE-01-06-02, University of Washington, 2002.

[72] R. Seker and M. M. Tanik. An information-theoretical framework for modeling component-based systems. *IEEE Trans. on Systems, Man and Cybernetics- Part C: Applications and Reviews*, 34(4), November 2004.

[73] S.El-Ansary. *Designs and Analyses in Structured Peer-to-Peer Systems.* PhD thesis, The Royal Institute of Technology (KTH), June 2005.

[74] Y. Shan and S. Kalyanaraman. Hybrid video downloading/streaming over peer-to-peer networks. In *Proceedings of the 4th IEEE ICME'2003 Multimedia and Expo*, volume 2, pages 665–668, July 2003.

[75] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423, July 1948.

[76] Y. Shen, Z. Liu, S. Panwar, K. W. Ross, and Y. Wang. Efficient substream encoding for p2p video on demand. [online-last visited on 31st of January,2008] on http://cis.poly.edu/ ross/papers/substreamEncodingP2P.pdf, November 2006. to be published.

[77] Y. Shen, Z. Liu, S. P. Panwar, K. W. Ross, and Y. Wang. Streaming layered encoded video using peers. In *[online-last visited on 31st of January,2008] on http://eeweb.poly.edu/P2P/*, 2006.

[78] Yanming Shen, Zhengye Liu, Shivendra S. Panwar, Keith W. Ross, and Yao Wang. Streaming layered encoded video using peers. In *IEEE International Conference on Multimedia and Expo (ICME)*, October 2005.

[79] J. Shneidman and D. Parkes. Rationality and self-interest in peer-to-peer networks. In *In Proc. of 2nd International Workshop on Peer-to-Peer Systems (IPTPS03)*, February 2003.

[80] K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *INFOCOM*, April 2003.

[81] I. Stoica, R. Morris, M. Kaashoek, and H. Balakrishnan. Chord: A scalable peer to peer lookup service for internet applications. In *In Proc. of ACM SIGCOMM01*, August 2001.

[82] Q. Sun, N. Daswani, and H. Garcia Molina. Maximizing remote work in flooding-based peer-to-peer systems. *Elsevier Computer Networks 48*, pages 1583–1598, 2005.

[83] S.R. Viola, S. Graf, and Kinshuk. Detecting learners' profiles based on the index of learning styles data. In *Proceedings of the International Workshop on Intelligent and Adaptive Web-based Educational Systems (IAWES 2007)*, pages 233–238, November 2007.

[84] A. L. Vitali, A. Borneo, M. Fumagalli, and R. Rinaldo. Video over ip using standard-compatible multiple description coding:an ietf proposal. *Journal of Zhejiang University SCIENCE A*, 7(5):668–676, 2006.

[85] S. Liu X. Bai, P. Zhang, and R. Kantola. Icn: Interest-based clustering network. In *International Conference on P2P Computing*, August 2004.

[86] L. Xiong and L. Liu. Peertrust: Supporting reputation based trust in peer-to-peer communuties. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):843–857, July 2004.

[87] X. Xu, Y. Wang, S. S. Panwar, and K. W. Ross. A peer-to-peer video-on-demand system using multiple description coding and server diversity. In

Internation Conf. Image Processing (ICIP'04), pages 1759–1762, October 2004.

[88] M. A. Yazıcı, K. Öztoprak, N.Akar, and G.B. Akar. Markov chain-based reliability modeling of multi-stream p2p video on demand systems. *submitted to IEEE Transactions on Reliability available on http://www.ceng.metu.edu.tr/k̃asim/papers/reliability.pdf*, 2007.

[89] R Zhang, A.R. Butt, and Y.C Hu. Topology-aware peer-to-peer on demand streaming. In *In Proceedings of 2005 IFIP Networking Conference (Networking)*, May 2005.

[90] X. Y. Zhang, Q.Zhang, Z. Zhang, Song G, and W. Zhu. A construction of locality-aware overlay network: moverlay and its performance. *IEEE Journal on Selected Areas in Communications*, 22(1):489–501, January 2004.

[91] B. Zhao, L. Huang, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz. Tapestry: A resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*, 22(1):41–53, 2004.

[92] M. Zhou and J. Liu. A hybrid overlay network for video on demand. In *IEEE International Conference on Communications (ICC 05)*, May 2005.

[93] M. Zink and A. Mauthe. P2p streaming using multiple description coded video. In *Proceedings of the 30th Euromicro Conference (Euromicro'04)*, pages 240–247, Rennes, France, 2004.

VITA

KASIM ÖZTOPRAK was born in Sivas in 1973. After completing his studies in the Malatya Science High School, in 1990, he entered to the Middle East Technical University, Turkey in 1991. He received the doctoral degree in February 2008, the masters degree in May 2000 and the bachelor degree in July 1996 in Computer Engineering from Middle East Technical University, Turkey.

While he was a Bachelor student, he started working for The Scientific and Technical Research council of Turkey (TUBITAK) as a member of the TR-NET project. After completion of bachelor degree, he is employed as a reserach assistant at Başkent University. Between 1997 and 2002, he is employed as a World Bank Consultant in the Project Coordination Unit of Ministry of Health. Since 2002, he is a member of Atılım University Computer Engineering Department as an Instructor.

During his professional studies, he was the Leader of Year 2000 (Y2K) Compliance Project for Health Sector in Turkey. He also consulted to Ministry of National Education for its Information Technology projects.

His research interests include computer networks, peer-to-peer systems, multimedia networking, computer architecture, and network security. He is a member of Institute of Electrical and Electronics Engineers (IEEE) and The Institute of Electronics, Information and Communication Engineers (IEICE).