



Faculty of Information And Communication Technology

**INCREASING SERVICE QUALITY OF MULTIMEDIA
STREAMING USING HYBRID PEER-TO-PEER MODEL**

AHMAD TAJUDDIN BIN SAMSUDIN

DOCTOR OF PHILOSOPHY

2010

**INCREASING SERVICE QUALITY OF MULTIMEDIA STREAMING
USING HYBRID PEER-TO-PEER MODEL**

AHMAD TAJUDDIN BIN SAMSUDIN

A thesis submitted

in fulfilment of the requirements for the degree of Doctor of Philosophy

Faculty of Information Technology and Communication

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2010

ABSTRACT

The peer-to-peer file sharing application has become increasingly popular for Internet users since the 1999 introduction of Napster. In the past decade, several more applications have been invented, such as Gnutella, FastTrack, Chord, Freenet, and BitTorrent. However, all of these applications use the downloading method that requires the source to be downloaded from one or more resources to one requesting peer. Now, alternative methods of file sharing delivery have been introduced; some examples include CELL, CollectCast, DAC, and GnuStream dan PAST, all of which use streaming to deliver media content to the user.

Because peer-to-peer file sharing and streaming systems - such as Video on Demand and IPTV - are popular among Internet users, developers and researchers have a tendency to combine the Video on Demand and peer-to-peer topology into one system. This way, the channel program of Video on Demand or IPTV can be inexpensively distributed on the Internet by utilizing the availability of peers; each peer becomes both a receiver and a supplier to another peer.

Many problems are associated with media streaming of file sharing applications, some of which include query-saturated networks, high latency in locating content, attempting to preserve uninterrupted streaming sessions, high peer load, flash crowds, and bottlenecks. As such, proposed system has been introduced in an effort to reduce the maintenance cost of overlay network topology, the routing and access costs of lookup services, and to service costs of streaming sessions. Proposed system is a combination of pure, client-server, and hierarchical peer-to-peer topology, and is categorized as a hybrid peer-to-peer business model.

Five experiments, consisting of four simulations and an expert evaluation, have been executed to determine the performance, scalability, maintenance, reliability and usability. The results reveal that proposed system can improve the quality of file sharing applications by reducing path length, peer load, and total usage while maintaining the overlay network topology in various churn rates, as well as locating a file in lookup services. The results also show that proposed system has sufficient scalability whenever the network size and number of queries increase.

ABSTRAK

Perisian perkongsian data berasaskan nod-kepada-nod amat popular di kalangan pengguna Internet sejak kemunculan Napster tahun. Kemudian pelbagai protokol dicipta seperti Gnutella, FastTrack, Chord, Freenet, and BitTorrent. Kesemua protokol diatas berasaskan penghantaran data secara muat-turun daripada satu atau lebih sumber kepada satu peminta. Selepas itu ia mula diganti dengan kaedah muat-aliran seperti CELL, CollectCast, DAC, GnuStream dan PAST.

Apabila sistem pengkongsian data nod-kepada-nod semakin popular dan pada masa yang sama permintaan sistem video atas permintaan dan IPTV turut meningkat maka wujud kecenderungan para pengkaji untuk membangunkan perisian yang membolehkan nod berkembang sebagai penerima dan penghantar. Eksplotasi terhadap topologi nod-kepada-nod membolehkan program TV dapat disebarikan melalui talian Internet secara meluas dengan kos rendah.

Didapati sistem pengkongsian data sedia ada masih ada beberapa masalah yang antara lain ialah ketepuan network akibat carian data yang meluas, masa menunggu yang tinggi semasa carian data dilakukan, sering kali sesi muat-lairan terganggu akibat kehilangan nod didalam sistem, beban kerja yang tinggi, capaian yang mendadak terhadap satu nod dan kepincangan tugas. Penyelidik mereka-bentuk sistem hibrid nod-kepada-nod dengan menggabungkan nod-kepada-nod tulin, pelayan-pelanggan dan hirarki dengan tutjuan untuk mengurangkan kos baik pulih topologi rangkaian, gelintar-carian dan perkhidmatan muat-aliran dapat dipertahankan.

Penyelidik telah mengadakan empat eksperimen melalui simulasi berkomputer di makmal, dan satu eksperimen berasaskan penilaian oleh para pengguna tegar sistem nod-kepada-nod. Hasil kajian menunjukkan bahawa sistem yang dicadangkan dapat meningkatkan kualiti perkhidmatan kepada perisian pengkongsian data secara muat-aliran dengan mengurangkan panjang-aliran, muatan-beban dan jumlah penggunaan mesej berbanding sistem nod-kepada-nod yang lain semasa proses carian data dan kekerapan keluar-masuk nod dalam rangkaian. Hasil kajian juga mendapati bahawa sistem memiliki keutuhan anjalan yang baik bila bilangan nod dan permintaan carian bertambah.

ACKNOWLEDGEMENT

In preparing this thesis, I was in contact with many people, researchers, academicians, and practitioners. They have contributed towards my understanding and thoughts. In particular, I wish to express my sincere appreciation to my thesis supervisor, Professor Dr. Nanna Suryana Herman, for encouragement, guidance, critics and friendship. Without his continued support and interest, this thesis would not have been the same as presented here.

I am also indebted to Telekom Research and Development (TMR&D) Malaysia for funding my Ph.D. study. Researchers and librarians at TMR&D and Fern-Universitat in Hagen of Germany also deserve special thanks for their assistance in supplying the relevant support and literatures.

My sincere appreciation also extends to all my colleagues and others who have provided assistant at various occasions. Their views and tips are useful indeed. I am grateful to all my family members.

DECLARATION

I declare that this thesis entitled “*Increasing service quality of multimedia streaming using hybrid peer-to-peer model*” is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :.....

Name :.....

Date :.....

TABLE OF CONTENT

	PAGE
ABSTRACT	ii
ABSTRAK	iii
DEDICATION	iv
ACKNOWLEDGEMENTS	v
DECLARATION	vi
LIST OF TABLES	xi
LIST OF FIGURES	xii
LIST OF ABBREVIATIONS	xiv
LIST OF SYMBOLS	xvi
LIST OF APPENDICES	xvii
CHAPTER	1
1 INTRODUCTION	
1.0 Introduction	1
1.1 Background	2
1.2 Problem statement	5
1.2.1 Traffic, peer load and latency	5
1.2.2 Preserve quality streaming session	6
1.2.3 Bottleneck and flash crowd	7
1.3 Research mission	8
1.4 Research questions	8
1.5 Research objective	9
1.6 Research scope	9
1.7 Research limitation	10
1.8 Research contributions	10
1.9 Conclusion	11
2 LITERATURE REVIEW	12
2.0 Introduction	12
2.1 Business model	16
2.1.1 Pure Peer-to-Peer	17
2.1.2 Hybrid Peer-to-Peer	18
2.1.3 Hierarchical Peer-to-Peer	20
2.2 Applications	21
2.2.1 File sharing	22
2.2.2 Peer-to-Peer Internet Protocol TV	24

CHAPTER	PAGE
2.2.3 Communication	25
2.2.4 Network and grid computing	27
2.2.5 Online gaming	28
2.3 Current application	29
2.4 Overlay network topology	29
2.4.1 Structured	31
2.4.2 Unstructured	41
2.5 Lookup service	50
2.5.1 Metadata	51
2.5.2 Routing	62
2.5.3 Delegation	68
2.5.4 Information retrieval	70
2.6 Heterogeneity	76
2.7 Streaming approach	81
2.8 Analysis summary	90
2.8.1 Churn	90
2.8.2 Bootstrap server mechanism	92
2.8.3 Lookup performance	93
2.8.4 Peer load	95
2.8.5 File availability, accessibility and lifetime	96
2.8.6 Scalability	100
2.8.7 Fairness	103
2.8.8 Intellectual property	105
2.8.9 Locality	108
2.8.10 Interoperability	112
2.9 Conclusion	113
3 RESEARCH METHODOLOGY	114
3.0 Introduction	114
3.1 Literature review	114
3.2 Research design	116
3.3 Experimental research design	116
3.4 Data collection	130
3.5 Result and analysis	131
3.6 Conclusion	131
4 RESEARCH DESIGN	132
4.0 Introduction	132
4.1 Overlay network topology	134
4.1.1 Distributed hash table	139
4.1.2 Reactive recovery	142
4.1.3 Periodic recovery	150
4.1.4 Adjustment operation	153
4.1.5 Redirect operation	154
4.1.6 Acting position	155
4.2 Lookup services	159
4.2.1 Central metadata index	160
4.2.2 Routing path	164

CHAPTER	PAGE
4.2.3 Database queries	174
4.2.4 Delegation	180
4.2.5 Presentation of result	182
4.3 Dynamic active and standby sender peer	183
4.3.1 Peer selection	184
4.3.2 Dynamic resources detection	185
4.3.3 Multi-source streaming approach	187
4.4 Ranking peer	189
4.4.1 Bandwidth	190
4.4.2 CPU power	192
4.4.3 Packet loss rate	193
4.4.4 Uptime	193
4.4.5 Memory	194
4.4.6 Storage	194
4.5 Conclusion	195
5 EXPERIMENTAL RESEARCH DESIGN	196
5.0 Introduction	196
5.1 P2PNetSim setup for Experiment A,B,C and E	196
5.2 Experiment A: Maintenance area	197
5.3 Experiment B: Access and Routing area	200
5.4 Experiment C: Access area	205
5.5 Experiment D: Access area	208
5.6 Experiment E: Service area	212
5.7 Data verification and validation	214
5.7.1 For Experiment A,B,C and E	214
5.7.2 For Experiment D	217
5.8 Data collection and dataset of statistical package	218
5.9 Conclusion	219
6 COMPARATIVE ANALYSIS RESULTS	220
6.0 Introduction	220
6.1 Sub-goal #1	220
6.1.1 Question 1	220
6.1.2 Question 2	222
6.1.3 Question 3	223
6.1.4 Question 4	227
6.2 Sub-goal #2	228
6.2.1 Question 5	228
6.2.2 Question 6	233
6.2.3 Question 7	236
6.2.4 Question 8	239
6.2.5 Question 9	242
6.2.6 Question 10	244
6.2.7 Question 11	245
6.2.8 Question 12	249
6.2.9 Question 13	251
6.2.10 Question 14	253

CHAPTER	PAGE
6.2.11 Question 15	256
6.2.12 Question 16	261
6.3 Sub-goal #3	265
6.3.1 Question 17	265
6.4 Sub-goal #4	268
6.4.1 Question 18	268
6.5 Conclusion	269
7 SUMMARY, CONCLUSION AND MMENDATIONS FOR FUTURE RESEARCH	270
7.0 Introduction	270
7.1 Summary	270
7.2 Conclusion	275
7.3 Future work	276
REFERENCES	280
APPENDICES	298

LIST OF TABLES

TABLE	TITLE	PAGE
2.0	Overlay network topology	31
2.1	Element in metadata record	51
2.2	Several studies by researchers on delegation	68
2.3	Key value is correct but the information about file is wrong	72
2.4	The way of P2P system present their result to peer	74
2.5	Path length in various P2P systems	94
3.0	Service quality of P2P system need to be measured	118
3.1	List of areas, service quality and data attribute measured in experiment	119
4.0	DHTs structures for child, leaf-child and leaf-peer table	140
4.1	Example of three tables of DHTs in the BELUMS network	141
4.2	To appoint an acting child supervisor	157
4.3	Metadata elements	161
4.4	To calculate point D	184
4.5	To calculate a new point D	186
4.6	Ranking peer based on their P_i	190
4.7	Scale point for bandwidth	192
4.8	Scale point for each CPU power	192
5.0	Simulation setting for Experiment A for each algorithm and scenario	198
5.1	Simulation settings for Experiment B for each algorithm	201
5.2	Simulation settings in Experiment C for each algorithm	206
5.3	Description of the seven modules implemented in experiment D	209
5.4	Simulation settings in Experiment E for each scenario	213
6.0	To calculate the range of total usage for BELUMS, Chord and Viceroy	225
6.1	To select the best algorithm for lookup services in term of path length	241
6.2	Tabulated of percentage of finding file in BELUMS with or without ACS in Experiment B	252
6.3	Number of completion streaming session in Experiment E	266
7.0	Maintenance quality in maintenance area for each P2P system	272
7.1	Performance and scalability quality in routing and access area	273
7.2	Usability quality in access area	273

LIST OF FIGURES

FIGURE	TITLE	PAGE
2.0	Pure Peer-to-Peer topology	18
2.1	Hybrid Peer-to-Peer topology	19
2.2	Hierarchical peer-to-peer topology	21
2.3	Delegation factor chosen by TMR&D researcher by frequency	69
2.4	Split-and-Merge Multicast	88
3.0	Research methodology using waterfall method	115
4.0	BELUMS overlay network topology	135
4.1	Structured overlay network for root-circle	136
4.2	Unstructured overlay network for cluster-circle and region-circle	137
4.3	RTT measurements for latency network	139
4.4	Process flow in $O(1)$ to $O(2)$ step searching via child supervisor	165
4.5	Process flow in $O(1)$ to $O(3)$ step searching via acting child supervisor	166
4.6	Ontology domains for video repository	177
4.7	Integration database queries and six rules for lookup services	181
6.0	Mean total usage by churn rate of BELUMS algorithm in Experiment A	221
6.1	Incoming and outgoing message at different churn rate of BELUMS in Experiment A	222
6.2	Mean total usage by maintenance algorithm in Experiment A	224
6.3	Mean total usage handled by peers of total usage at different churn rate in Experiment A	226
6.4	Mean total usage handled by peers comparison between BELUMS, Chord and Viceroy at different churn rate in Experiment A	226
6.5	Mean peer load by maintenance algorithm in Experiment A	227
6.6	Mean path length by lookup service in 8192 BELUMS's network topology	230
6.7	Mean path length by lookup service in 16384 BELUMS's network topology	231
6.8	Mean path length comparison between BELUMS' with or without ACS in Experiment B	232
6.9	Mean total usage by lookup service algorithm in 8192 BELUMS network topology	234

FIGURE	TITLE	PAGE
6.10	Mean total usage by lookup service algorithm in 16384 BELUMS network topology	235
6.11	Mean total usage handled by peer comparison between BELUMS with or without ACS in Experiment B	235
6.12	Mean peer load against query by lookup service algorithm in BELUMS network topology	237
6.13	Mean peer load against query in BELUMS' Plan 4 and 5	238
6.14	Mean path length by lookup service algorithms in Experiment B	240
6.15	Mean total usage by lookup service algorithms in Experiment B	243
6.16	Mean peer load against query by lookup service algorithms in Experiment B	244
6.17	Mean difference of path length between 8192 and 16384 populations in Experiment B	247
6.18	Mean difference of total usage between 8192 and 16384 populations in Experiment B	247
6.19	Mean difference of peer load between 8192 and 16384 populations in Experiment B	248
6.20	Cumulative of number of peers in five time of simulation to find and share the rare item in Experiment C	250
6.21	Trend of finding file in BELUMS (8192 peers) with different number of child supervisor in Experiment B	252
6.22	Percent chosen by number of record in Experiment D	254
6.23	Percentage of satisfaction scale by method in Experiment D	254
6.24	Percentage of satisfaction scale by method in Experiment D	255
6.25	Mean rating by method in Experiment D	255
6.26	Mean rating by method in Experiment D	255
6.27	Mean rating by method in Experiment D	258
6.28	Percent by method in Experiment D	258
6.29	Percent by method in Experiment D	259
6.30	Percent by method in Experiment D	259
6.31	Is it adding the file format selection is complicated?	259
6.32	Mean rating by method in Experiment D	260
6.33	Mean rating by method in Experiment D	260
6.34	Percent by group B1 and B2 in Experiment D	260
6.35	Percent by elements of delegation in Experiment D	262
6.36	Mean rating by method (with or without Six Rules) in Experiment D	262
6.37	Percent of method chosen by expert in Experiment D	263
6.38	Percent of method chosen by expert in Experiment D	263
6.39	Percent of method chosen by expert in Experiment D	263
6.40	Percent of method chosen by expert in Experiment D	264
6.41	Packet loss rate observed in Experiment E	267
6.42	Percent of method chosen by expert in Experiment D	268

LIST OF ABBREVIATIONS

ACS	Acting child supervisor
ANSI	American National Standards Institute
API	Application Programming Interface
ARPA	Advanced Research Projects Agency
ARPANET	Advanced Research Projects Agency Network
ASQC	American Society for Quality Control
ASCII	American Standard Code for Information Interchange
ATA	Approximate Text Addressing
BELUMS	Bandwidth, cEntral processing unit power, packet Loss rate, Uptimes, Memory, and Storage
CAN	Content Addressable Network
CMMI	Capability Maturity Model Integration
CPU	Central Processing Power
CS	Child supervisor
DAC _{p2p}	distributed differentiated admission control protocol
DARPA	Defense Advanced Research Projects Agency
DHT	Distributed Hash Table
DLM	Dynamic Layer Management
DNS	Domain Name System
DRAGON	Dynamic Resource Allocation in GMPLS Optical Networks
DRM	Digital rights management
GB	Gigabytes
GHz	Gigahertz
GMPLS	Generalized Multiprotocol Label Switching
GUI	Graphical User Interface
HADAS	Heterogeneity-Aware Distributed Access Structure
HCR	High Churn Rate
ICT	Information and Communication Technology
IP	Internet Protocol
IPTV	Internet Protocol Television
ISO	International Organization for Standardization
JDK	Java Development Kit
JRE	Java Runtime Environment
Kbps	kilobit per second
LCR	Low Churn Rate
LRU	Least Recently Used
LST	Lightweight SuperPeer Topologies
MB	Megabytes
Mbps	megabit per second
MCR	Moderate Churn Rate
MHz	Megahertz

LIST OF ABBREVIATIONS

MP3	Moving Picture Experts Group-1 Audio Layer 3
MSN	The Microsoft Network
NS	Nucleus supervisor
ODIN	Optical Dynamic Intelligent network
ONT	Overlay network topology
OSI	Open Systems Interconnection
OTS _{p2p}	optimal media data assignment algorithm
P2P	Peer-to-Peer
RAM	Random Access Memory
RFID	Radio-Frequency Identification
RTP	Real Time Protocol
RTT	Round-trip time
SHA	Secure Hash Algorithm
SIAM	Security, Intelligent Application and Multimedia
SWAM-V	Voronoid-based Small-World Access Methods
TCP	Transmission Control Protocol
Test ID	Number of Test
TMR&D	Telekom Research and Development Sdn. Bhd.
TTL	Time-to-live
TV	Television
UDP	User Datagram Protocol
UGC	User Generated Content
VOIP	Voice over Internet Protocol
XOR	Exclusive-OR
XML	eXtensible Markup Language

LIST OF SYMBOLS

A_i	access cost
B	bandwidth
b	number of entries
ϵ	central processing power
C_i	total cost
i	the set of numbers
ID	identity
l	level or height
$\log N$	logarithm for base 2.
M_i	maintenance cost
N, n	number of peers
O	big O or Landau notation
P_i	scale of rank from 0 to 100
P_R	ranking peer
P_L	serving limitation
Ptr	pointer list
P_S	storage of peer
P_{LR}	packet loss rate
P_u	uptime of peer
R_a	available bandwidth
R_i	routing cost
R_o	upstream or upload or offer rate
S_a	active sender
S_i	service cost
S_s	standby sender
w	weightage
x	displacement
U	uptime
M	memory

LIST OF APPENDICIES

APPENDIX	TITLE	PAGE
A	Percentages of aggregated P2P file sharing traffic in Internet2 network	298
B	Example of P2P application	299
C1	Example of structured overlay network topology	304
C2	Example of unstructured overlay network topology	308
D	Example of graphical user interface for result presentation	310
E	Streaming approach	312
F	Summary of heterogeneities were implemented in P2P system	313
G	List of sub-goal, quantifiable questions and related indicators	314
H	Selected other designs in experiment	317
I	Variable and parameter setup used in Experiment	318
J	Description of the 50 methods implemented in Experiment D	320
K1	Join, leave, and periodic message events setup in Experiment A	324
K2	Initial value for rank, limit and upstream for each 20 peers in Experiment E	326
K3	Sample of expected total usage for BELUMS, Chord and Viceroy	327
L1	Sample of output log file for Experiment A	328
L2	Sample of output log file for Experiment B	329
L3	Sample of output log file for Experiment C	330
L4	Sample of output log file for Experiment E	331
M	Relationship between research objective, sub-goal, research question, and problem statement	332
N	Summary of proposed design	333

CHAPTER 1

INTRODUCTION

1.0 Introduction

File sharing becomes trendier as peer-to-peer applications continue to emerge and provide options for the community to share their content, such as KaZaA, BitTorrent, eDonkey2000, Napster, Gnutella and Freenet. Nowadays P2P technology is used not only for downloading but also for streaming. Researchers have been trying to implement media streaming in a P2P environment, such as CELL, CollectCast, SplitStream, ZIGZAG, PeerStream and GnuStream.

This research study focuses on streaming concepts instead of downloading for file-sharing applications. There is a difference between downloading and streaming techniques, as referred to anon. (2006). First, in streaming, the media plays while it is being transferred; whereas in downloading, files from the peer's storage centre are stored directly onto a local storage disk. So, in a download-based system, the user does not need to be ready to watch a video, whereas in a streaming-based system, the user should be prepared to watch it once the content has been transferred.

Secondly, in a streaming environment, the user can determine if the media is what the user wants after transferring only a few seconds of it, whereas in the download method the user can not do that. Furthermore, it can be disappointing to discover that the content is not right after eight days of a download process, as can happen with a particularly large media file.

Last but not least, the download environment allows users to have a local copy to modify, store and copy to different locations, or to share with others, whereas in streaming an owner can protect the copyright. In this proposed design a streaming concept instead of a download concept has been introduced, but the platform still can be used for other P2P applications.

1.1 Background

P2P systems can be based on an overlay network topology, which is structured, as well as on unstructured systems. Keong *et al.* (2005) and Chawathe *et al.* (2003) defined the structured overlay network topology as tightly controlled and characterised routing the query message to overlay paths as easy, whereas the unstructured system is composed of peers joining and leaving the network with some loose rules, but without any prior knowledge of the topology.

In both structured and unstructured P2P overlay network topologies, there are different possibilities of guarantee to locate data. Gnutella cannot guarantee its ability to locate data, although it uses flooding. Gnutella requires $\theta(N)$ steps with a limited TTL, whereas the centralized servers of Napster, BitTorrent, SPON, and eDonkey2000 should, in theory, be able to provide the location of data. Meanwhile, the majority of structured P2P lookup services offer DHT, in which DHT can guarantee data location within their overlay network topology. The application translates the exact name of the file into a key and performs a query to match a key over DHT routing information; in example, Chord and Viceroy can locate data in small $O(\log N)$, where N is the number of peers in the network.

Napster is a pioneer in introducing a centralized server. The server will collect all meta-data and provide a location of data to peers. SPON, proposed by Riley and Scheideler (2004a), and Kothapalli and Scheideler (2005), is a tree-based network that manages group updates and supports efficient broadcasting. SPON uses a supervisor peer to maintain the network during node arrivals and departures and routes broadcasts using direct connections between nodes.

BitTorrent has a meta-data collector server, which stores *.torrent* files, as well as a tracker server, which keeps track of the peer activity (in example, download and upload) and availability of the pieces of files. In this P2P system, the response time of lookup services is constant, and the resulting set of queries are in multiple files and peers (online and offline). Unfortunately, the BitTorrent server does not take care of any contents deleted from the system. In other words, the central server still stores the *.torrent* file, although that particular file does not exist anymore in the network.

It is no wonder that eDonkey2000 also implements the idea of a centralized server. Unfortunately, on September 28, 2005, eDonkey2000 officially closed its doors due to illegal downloading. eDonkey2000 received a cease-and-desist letter from the Recording Industry Association of America as a result of the June 2001 Supreme Court ruling that makers of software that facilitates copyright infringement are liable for that infringement (Veiga, 2006).

The P2P researchers believe that, in order to be a success in P2P applications, systems need to protect the content, the content owner, the peer itself and also sharing activities. Researchers should consider implementing the appropriate protection mechanism into P2P file-sharing applications.

Many previous studies show that researchers tried to take advantage of the different characteristics of peers such as offer rate, available bandwidth, buffer, storage, memory, uptime, etc., to offer better quality of downloading or streaming process.

Gnutella proposed Ultrapeer topology, in which the peer itself becomes a server. Ultrapeers perform extra work, however, compared to a normal peer. They have to receive multiple meta-data and query requests from connected peers. To become an ultrapeer, the peer should be able to meet the constraints; not be firewalled, and have a suitable operating system as well as sufficient bandwidth, uptime, RAM and CPU speed.

KaAza also introduced a similar ultrapeer called the SuperPeers. The difference is that the peers can nominate themselves as SuperPeers, although their connection speed is 128kbps or the CPU speed is 445 MHz! Kleis *et al.* (2005a; 2005b) propose LST which is designed to have lower complexity and management overhead than structured P2P overlay network topology. Another study conducted by Xiao *et al.* (2005) shows that the quality of a SuperPeer system is significantly improved under the DLM algorithm. In DLM, every peer decides to be a SuperPeer or leaf-peer independently without global knowledge.

There are a few systems proposed that use multicast technology to support streaming to a large number of receivers simultaneously, and their intention is to stream from one single-source to many peers. This technique can be used to improve the quality of media streaming, such as IPTV and video on demand. Xu *et al.* (2002), ZIGZAG (Tran *et al.*, 2003), GridMedia (Zhang *et al.*, 2005), and Split-Merge (Kulkarni and Markham, 2005) are using peer-to-peer topology, in which these systems utilize the existence of nodes to receive and distribute the packet stream to other nodes. ZIGZAG is concerned about the unpredictable nature of peer behaviours and resource limitations at the receiver peer. In Split-Merge, they take into consideration the upstream capacity.

Cai *et al.* (2005) proposes CELL mechanism to utilize the peer buffers, in which the supplying and requesting peer will determine whether the requesting peer should be able to cache some of the segmented video files. CELL uses Gnutella-like lookup services and stops the query once the system finds one caching host. Freenet (Clarke *et al.*, 2000) also implemented caching; if a file is found during lookup

service, and the file is successfully retrieved by the original requester, then the file will cache on sequence upstream requester.

In CollectCast applications, the Hefeeda *et al.* (2003) use offer rate and available bandwidth as an indicator of peer goodness. They utilise modified Tapestry lookup services, and return one or more supplier peer. Then, CollectCast does a peer selection based on offer rate and available bandwidth over topology-aware selection techniques. Thus, two groups of peers will be created: active sender and standby sender. CollectCast dynamically switches active senders and standby senders, so that the collective network performance out of the active senders remains satisfactory.

In this part of the research, the researcher focuses on the distributed and supervised peer, how to rank the peers, file delegation and dynamic standby peers. The proposed design uses the advantages of the server-client business model to search and retrieve information. To make sure there is no flash crowd or excessive access by users, the proposed design needs to improve the delegation policy. To provide a dynamic standby peer, the system uses a content hash key value as an indicator to search for an identical content in the P2P network. Proposed designs consider the selection of the best peer to serve a streaming session as a crucial step in providing a good quality video stream.

1.2 Problem statement

1.2.1 Traffic, peer load and latency

The main problem of pure P2P systems is to assign and locate resources among peers. P2P systems differ in the way in which they construct the overlay topology and distribute queries from node to node. The advantage is that they can easily accommodate a highly transient node population. The disadvantage is that it is hard to find the desired files without distributing queries widely.

In an unstructured and pure P2P, when one node joins or leaves the network, does it have to alert the other peers about its activities? If not, how do the other peers know this peer is in their group?

If one closely looks into the majority of structured and pure P2Ps, system provide DHT, from which a much better lookup service is provided within their overlay network topology, whereas in hybrid P2P, the challenge is to figure out the mechanism for organizing the peers that allows them to cooperate, provide a metadata index and, finally, enable any peer to resourcefully locate contents.

However, structured designs are likely to be less resilient in the face of a very transient user population, precisely because it is hard to maintain the structure of overlay network topology required for routing to function efficiently when nodes are joining and leaving at a high rate.

Kothapalli and Scheideler (2005) said that server/client-based systems can provide guarantees and are, therefore, preferable for critical applications that need a high level of reliability; but the peer-to-peer systems can scale to millions of sites with low-cost hardware whereas the classical approach of using server-client systems does not scale well unless powerful servers are provided. BitTorrent is one of an example of the server-client search approach, where the latency of lookup services is constant and the peer load is minimal. Unfortunately, BitTorrent does not take care of any deleted contents from the system or the status of the peer itself. The server also seems to work alone and not to communicate or replicate with other servers.

1.2.2 Preserve quality streaming session

The majority of P2P applications do not deploy a proper delegation process. For example, in BitTorrent, every peer has a right to search for particular data in P2P networks. The requesting peer has a right to get all results without any filtering. If

that peer shares a popular file, then the owner will face a problem and his computer will always be connected with other computers. As a consequence, his available bandwidth will drop, peer load will increase and, finally, it will not be able to preserve the quality of media while streaming.

CollectCast does a peer selection based on the offer rate and available bandwidth over topology-aware selection technique. Thus, two groups of peers; for example, active and standby senders will be performed. System dynamically switches active senders and standby senders, so that the collective network performance out of the active senders remains satisfactory. However, the standby sender in CollectCast is static. If a new peer joins the network and has that file, it will not take part as a standby sender.

1.2.3 Bottleneck and flash crowd

If a server in a hybrid P2P system is down due to maintenance or bottleneck, the system faces the problem of assigning and locating the resources. This problem also occurs in a server/client-based system. Sometimes, when a thousand clients rapidly try to connect to the server for media downloading, the flash crowd symptom always arises (in example, the CNN portal on September 11, 2001).

Actually, pure P2P also exhibit the flash crowd when without a proper delegation, for example a thousand requesting peers try to download a popular file from the same source. Sometime it also occurs at node within the chain of lookup service especially system used uni-path random walk such as Freenet.