

The copyright © of this thesis belongs to its rightful author and/or other copyright owner. Copies can be accessed and downloaded for non-commercial or learning purposes without any charge and permission. The thesis cannot be reproduced or quoted as a whole without the permission from its rightful owner. No alteration or changes in format is allowed without permission from its rightful owner.



**A NEW COORDINATED CACHING PLACEMENT MECHANISM  
FOR VOD IN NAMED DATA NETWORKING**



**RASHA SALEEM ABBAS**

**UUM**  
Universiti Utara Malaysia

**DOCTOR OF PHILOSOPHY  
UNIVERSITI UTARA MALAYSIA  
2025**



Awang Had Salleh  
Graduate School  
of Arts And Sciences

Universiti Utara Malaysia

**PERAKUAN KERJA TESIS / DISERTASI**  
(*Certification of thesis / dissertation*)

Kami, yang bertandatangan, memperakukan bahawa  
(*We, the undersigned, certify that*)

**RASHA SALEEM ABBAS**

calon untuk Ijazah  
(*candidate for the degree of*)

**PhD**

telah mengemukakan tesis / disertasi yang bertajuk:  
(*has presented his/her thesis / dissertation of the following title*):

**"A NEW COORDINATED CACHING PLACEMENT MECHANISM  
FOR VOD IN NAMED DATA NETWORKING"**

seperti yang tercatat di muka surat tajuk dan kulit tesis / disertasi.  
(*as it appears on the title page and front cover of the thesis / dissertation*).

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : **21 January 2025**.

*That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on:*  
**21 January 2025.**

Pengerusi Viva:  
(*Chairman for VIVA*)

**Assoc. Prof. Dr. Massudi Mahmuddin**

Tandatangan  
(*Signature*)

Pemeriksa Luar:  
(*External Examiner*)

**Assoc. Prof. Dr. Rosilah Hassan**

Tandatangan  
(*Signature*)

Pemeriksa Dalam:  
(*Internal Examiner*)

**Dr. Ahmad Suki Che Mohamed**

Tandatangan  
(*Signature*)

Nama Penyelia/Penyelia-penyelia:  
(*Name of Supervisor/Supervisors*)

**Ts. Dr. Shahrudin Awang Nor**

Tandatangan  
(*Signature*)

Nama Penyelia/Penyelia-penyelia:  
(*Name of Supervisor/Supervisors*)

**Dr. Amran Ahmad**

Tandatangan  
(*Signature*)

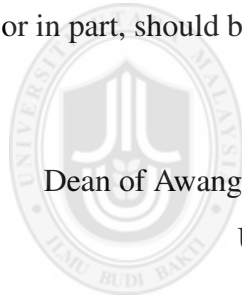
Tarikh:

(*Date*) **21 January 2025**

## **Permission to Use**

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:



Dean of Awang Had Salleh Graduate School of Arts and Sciences

UUM College of Arts and Sciences

Universiti Utara Malaysia

06010 UUM Sintok

## Abstrak

Walaupun terdapat kemajuan ketara dalam Rangkaian Data Dinamakan (NDN), pengendalian trafik Video atas Permintaan (VoD) yang cekap masih menjadi cabaran akibat strategi caching yang tidak optimum. Hal ini menyebabkan peningkatan kelewatan hujung ke hujung (E2EDelay) dan beban pelayan yang tinggi. Mekanisme sedia ada sering gagal mengurus permintaan video dalam talian serentak, terutamanya dalam persekitaran rangkaian yang dinamik. Kajian ini menangani ketidakefisienan tersebut dengan mencadangkan satu mekanisme penempatan caching terkoordinasi baharu yang direka khas untuk perkhidmatan VoD dalam persekitaran NDN. Objektif pertama kajian ini adalah menilai ciri fungsian tujuh mekanisme penempatan caching sedia ada melalui simulasi, bagi mengenal pasti keperluan fungsi penting untuk penghantaran kandungan VoD yang optimum. Berdasarkan dapatan ini, mekanisme baharu iaitu Coordination Caching Placement by Video Partitioning (CPVP), diperkenalkan. CPVP bertujuan meningkatkan kecekapan caching, mengurangkan kependaman, dan menambah baik pengalaman pengguna. CPVP menggunakan teori hipergraph untuk memodelkan tingkah laku caching koperatif dalam rangkaian. Ia membahagikan kandungan video secara dinamik dan menyimpan segmen video di sepanjang laluan ke klien melalui tiga penghala yang berdekatan. Mekanisme ini merangkumi tiga skema utama: Video Aggregation Scheme (VAS) untuk mengenal pasti video penuh dan menentukan penghala yang boleh digunakan untuk caching; Video Cache Decision Scheme (VCDS) yang mengurangkan redundansi dan kependaman; serta Video Partitioning Scheme (VPS) yang membahagikan video kepada tiga blok untuk disimpan secara teragih. Simulasi dijalankan menggunakan ndnSIM 2.0 ke atas empat topologi rangkaian, menilai empat metrik prestasi utama. Hasil kajian menunjukkan bahawa CPVP mengatasi mekanisme sedia ada seperti Leave Copy Everywhere (LCE), Leave Copy Down (LCD), dan caching secara kebarangkalian (Prob). Secara khususnya, CPVP mencatatkan pengurangan 22.3% dalam E2EDelay purata dan peningkatan 56.5% dalam nisbah hit cache pada topologi GEANT. Kajian ini menyumbang kepada pembangunan strategi caching yang lebih cekap dalam NDN dan menawarkan peningkatan ketara dalam kualiti penstriman VoD dan prestasi rangkaian.

**Kata Kunci:** Skim Pengagregatan Video (VAS), Skim Keputusan Cache Video (VCDS), Skim Pembahagian Video (VPS), Hipergraf, Penempatan Cache Penyelarasan oleh Pemisahan Video (CPVP).

## Abstract

Despite significant advancements in Named Data Networking (NDN), the efficient handling of Video on Demand (VoD) traffic remains a challenge due to suboptimal caching strategies, resulting in increased End-to-End Delay (E2EDelay) and elevated server load. Existing mechanisms often fail to manage simultaneous online video requests effectively, particularly under dynamic network conditions. This study addresses these inefficiencies by proposing a novel coordinated caching placement mechanism tailored for VoD services in NDN environments. The first objective of the research is to evaluate the functional characteristics of seven existing caching placement mechanisms through simulation, identifying the features essential for optimal VoD content delivery. Building on these insights, a new mechanism, Coordination Caching Placement by Video Partitioning (CPVP), is introduced. CPVP aims to enhance caching efficiency, reduce latency, and improve user experience. CPVP utilises hypergraph theory to model cooperative caching behaviour across the network. It dynamically partitions video content and strategically stores segments along the path to clients using three neighbouring routers. The mechanism incorporates three integrated schemes: the Video Aggregation Scheme (VAS), which identifies the full video and determines cacheable routers; the Video Cache Decision Scheme (VCDS), which minimises redundancy and latency; and the Video Partitioning Scheme (VPS), which divides the video into three blocks for distributed caching. Simulations are conducted using ndnSIM 2.0 across four network topologies, evaluating four key performance metrics. Results demonstrate that CPVP outperforms existing mechanisms such as Leave Copy Everywhere (LCE), Leave Copy Down (LCD), and probabilistic caching (Prob). Specifically, CPVP achieves a 22.3% reduction in average E2EDelay and a 56.5% improvement in cache hit ratio on the GEANT topology. This research contributes to the development of efficient caching strategies in NDN and offers practical improvements in VoD streaming quality and network performance.

**Keywords:** Video Aggregation Scheme (VAS), Video Cache Decision Scheme (VCDS), Video Partitioning Scheme (VPS), Hypergraph, Coordination Caching Placement by Video Partitioning (CPVP).

## Acknowledgements

In the name of ALLAH, Most Gracious, Most Merciful:

*“Work so that Allah will see your work and (so will) his messenger and the believers;”*

(The Holy Quran – AtTawbah 9:105)

Conducting this research marks the end of an exciting and eventful journey. The completion of this dissertation signifies the fulfilment of a long-awaited goal. Without the professional, academic, and personal support of the following excellent and talented people, I could not have achieved this. I will start by extending my deep and sincere gratitude to my two supervisors, AP. Ts Dr. Shahrudin Awang Nor, and Dr. Amran Ahmad (School of Computing, Universiti Utara Malaysia), for their tireless encouragement, wisdom, and experience, who provided me with constant guidance and constructive criticism throughout my research. May Allah give them their reward and make them happy. In addition to my never-ending dedication to my work, I must say a huge thanks to the current and past members of InterNetWorks Research Lab, with whom I enjoyed working. I would like to thank the Internet Society’s Next Generation Leaders (NGL) programme while serving as an ISOC fellow for the Internet Engineering Task Force (IETF). Finally, my heartiest gratitude goes to my family... to my late father... and my mother... to my beloved husband Sadaq for his understanding, support, and love... to my clever and loyal sons Jaafer, Osamah, and Abhaj... and my fantastic and lovely daughter Tabarek... and to my dearest and soul sister who always has faith in me and prays for my success.

## Declaration Associated with This Thesis

Some of the works presented in this thesis have been published or submitted as listed below:

[1] Rasha Saleem Abbas, Shahrudin Awang Nor, and Amran Ahmad. “Coordination and Non-coordination Caching Placement Algorithms in Named Data Network for Video on Demand Services,” *Journal of Computer Engineering (IOSR-JCE)*, 24(2), pp. 14–21, Sept. 2022.

[2] Rasha Saleem Abbas, Shahrudin Awang Nor, and Amran Ahmad. “A Review of Cache Placement Algorithms for Video on Demand in Named Data Networking,” *Journal of Advanced Research in Dynamical and Control Systems (JARDCS)*, pp. 1322–1332, Sept. 2018.

[3] Rasha Saleem Abbas, Shahrudin Awang Nor, and Amran Ahmad. “A Study of Cache Placement Algorithms for Video on Demand in Named Data Networking,” Technical Report, InterNetWorks Research Laboratory, School of Computing, Universiti Utara Malaysia, 2017

# Contents

Permission to Use . . . . .	i
Abstrak . . . . .	ii
Abstract . . . . .	iii
Acknowledgements . . . . .	iv
Contents . . . . .	v
List of Tables . . . . .	x
List of Figures . . . . .	xii
List of Abbreviations . . . . .	xvii
<b>CHAPTER ONE INTRODUCTION . . . . .</b>	<b>1</b>
1.1 Background of the Study . . . . .	3
1.2 Video on Demand in NDN . . . . .	6
1.3 Research Motivation . . . . .	10
1.4 Problem Statement . . . . .	13
1.5 Research Questions . . . . .	16
1.6 Research Objectives . . . . .	17
1.7 Research Scope . . . . .	18
1.8 Significance of the Research . . . . .	19
1.9 Research Contributions . . . . .	21
1.10 Thesis Outline . . . . .	22
<b>CHAPTER TWO LITERATURE REVIEW . . . . .</b>	<b>24</b>
2.1 NDN Classification . . . . .	25
2.1.1 NDN System Architecture . . . . .	26
2.1.2 NDN System Services . . . . .	27
2.1.2.1 Forwarding . . . . .	27
2.1.2.2 Routing . . . . .	27
2.1.2.3 Security, Privacy & Trust . . . . .	28
2.1.2.4 Mobility . . . . .	28
2.1.2.5 Caching . . . . .	29
2.1.3 Applications . . . . .	30

2.2	Classification of Caching Mechanisms . . . . .	33
2.2.1	Cache Replacement . . . . .	34
2.2.2	Cache Placement . . . . .	35
2.3	In-network Caching . . . . .	38
2.3.1	Off-path Caching . . . . .	40
2.3.2	On-path caching . . . . .	41
2.3.2.1	Non-coordinated Caching Policy . . . . .	42
2.3.2.2	Coordinated Caching Policy . . . . .	43
2.4	Cache Placement Mechanisms and VoD Services . . . . .	46
2.4.1	Leave Copy Everywhere (LCE) . . . . .	47
2.4.2	Leave Copy Down (LCD) . . . . .	50
2.4.3	Random Choice Caching (Rand) . . . . .	54
2.4.4	Probabilistic Cache (Prob) . . . . .	56
2.4.5	Path Probabilistic Cache (PProb) . . . . .	58
2.4.6	Hybrid Caching (Cross) . . . . .	60
2.4.7	Centrality-based Mechanism . . . . .	60
2.4.8	A Cache Placement Based on Compound Popularity Mechanism (CBCP) . . . . .	63
2.4.9	A Cache Placement Mechanism Based on Entropy Weighting Method and TOPSIS (CBEWT) . . . . .	64
2.4.10	Coordination Caching Placement for VoD in NDN . . . . .	66
2.5	Comparison of Caching Placement Mechanisms . . . . .	69
2.6	Summary . . . . .	71
<b>CHAPTER THREE RESEARCH METHODOLOGY . . . . .</b>		<b>73</b>
3.1	Research Operational Framework . . . . .	74
3.2	The Proposed Conceptual Model of Coordination Placement of Video Partitioning mechanism (CPVP) . . . . .	79
3.3	Verification and Validation Model . . . . .	81
3.4	Evaluation . . . . .	83
3.4.1	Simulation Construction . . . . .	85
3.4.1.1	Simulation Model . . . . .	87
3.4.1.2	Simulation Scenario . . . . .	90

3.4.1.3	Network Topology . . . . .	90
3.4.1.4	Content Popularity . . . . .	94
3.4.1.5	Cache Size . . . . .	95
3.4.1.6	The catalogue size . . . . .	96
3.4.1.7	Simulation Parameter Settings . . . . .	96
3.4.2	Simulation Setup . . . . .	98
3.4.2.1	The Model of Video . . . . .	99
3.4.2.2	Performance Metrics . . . . .	101
3.4.3	Confidence Level of Simulation Results . . . . .	109
3.5	Summary . . . . .	109

## CHAPTER FOUR THE EMPIRICAL EVALUATION OF CACHING

	<b>PLACEMENT MECHANISMS . . . . .</b>	<b>111</b>
4.1	Introduction . . . . .	111
4.2	Configuration of the Mechanisms . . . . .	112
4.3	Simulation Scenario . . . . .	115
4.4	Simulation Results and Evaluation of Seven Placement Mechanisms . . .	116
4.4.1	Average Hit Ratio . . . . .	117
4.4.1.1	Average Hit Ratio Using 1 GB and $\alpha = 0.65$ . . . . .	117
4.4.1.2	Average Hit Ratio Using 10 GB and $\alpha = 0.65$ . . . . .	118
4.4.1.3	Average Hit Ratio Using 100 GB and $\alpha = 0.65$ . . . . .	119
4.4.1.4	Average Hit Ratio Using 1 TB and $\alpha = 0.65$ . . . . .	120
4.4.1.5	Average Hit Ratio Using 1 GB to 1 TB and $\alpha = 0.65$ . . . . .	121
4.4.1.6	Average Hit Ratio Using 1 GB to 1 TB and $\alpha = 2.0$ . . . . .	123
4.4.2	Average E2EDelay . . . . .	124
4.4.2.1	Average E2EDelay Using 1 GB and $\alpha = 0.65$ . . . . .	124
4.4.2.2	Average E2EDelay Using 10 GB and $\alpha = 0.65$ . . . . .	125
4.4.2.3	Average E2EDelay Using 100 GB and $\alpha = 0.65$ . . . . .	126
4.4.2.4	Average E2EDelay Using 1 TB and $\alpha = 0.65$ . . . . .	127
4.4.2.5	Average E2EDelay Using 1 GB to 1 TB and $\alpha = 0.65$ . . . . .	128
4.4.2.6	Average E2EDelay Using 1 GB to 1 TB and $\alpha = 2.0$ . . . . .	129
4.4.3	Server Load Reduction . . . . .	130
4.4.3.1	Server Load Reduction Using 1 GB and $\alpha = 0.65$ . . . . .	130

4.4.3.2	Server Load Reduction Using 10 GB and $\alpha = 0.65$ . . . .	131
4.4.3.3	Server Load Reduction Using 100 GB and $\alpha = 0.65$ . . .	132
4.4.3.4	Server Load Reduction Using 1 TB and $\alpha = 0.65$ . . . . .	132
4.4.3.5	Server Load Reduction Using 1 GB to 1 TB and $\alpha = 0.65$	133
4.4.3.6	Server Load Reduction Using 1 GB to 1 TB and $\alpha = 2.0$	135
4.4.4	Reduction in the Network Footprint . . . . .	136
4.4.4.1	Total Reduction in Network Footprint Using 1 GB and $\alpha = 0.65$ . . . . .	136
4.4.4.2	Total Reduction in Network Footprint Using 10 GB and $\alpha = 0.65$ . . . . .	137
4.4.4.3	Total Reduction in Network Footprint Using 100 GB and $\alpha = 0.65$ . . . . .	138
4.4.4.4	Total Reduction in Network Footprint Using 1 TB and $\alpha = 0.65$ . . . . .	138
4.4.4.5	Total Reduction in the Network Footprint Using 1 GB to 1 TB and $\alpha = 0.65$ . . . . .	139
4.4.4.6	Total Reduction in the Network Footprint Using 1 GB to 1 TB and $\alpha = 2.0$ . . . . .	141
4.4.5	Functional Requirements of VoD in NDN . . . . .	142
4.5	Summary . . . . .	143

## **CHAPTER FIVE DESIGN COORDINATION CACHING PLACEMENT**

	<b>BY VIDEO PARTITIONING (CPVP) . . . . .</b>	<b>145</b>
5.1	Introduction . . . . .	145
5.2	Network Model . . . . .	147
5.3	Theoretical Analysis . . . . .	152
5.3.1	Analysis of Modeling Content Placement Problem . . . . .	154
5.3.2	Analysis of Coordinated and non-coordinated Caching Mechanism	157
5.4	Coordinated Caching Placement by Video Partitioning (CPVP) . . . . .	163
5.4.1	Create Cooperative Tables . . . . .	168
5.4.2	Video Aggregation Scheme (VAS) . . . . .	175
5.4.3	Video Caching Decision Scheme (VCDS) . . . . .	179
5.4.4	Video Partitioning Scheme (VPS) . . . . .	181

5.4.4.1	Description of VPS . . . . .	182
5.4.4.2	VPS Mechanism & Flowchart . . . . .	186
5.5	Verification and Validation of Coordination Caching Placement by Video Partitioning Mechanism (CPVP) . . . . .	194
5.6	Summary . . . . .	196
<b>CHAPTER SIX SIMULATION RESULTS AND EVALUATION . . . . .</b>		<b>198</b>
6.1	Simulation Environment . . . . .	198
6.1.1	Simulation Setup . . . . .	199
6.2	Simulation Results and Discussion . . . . .	200
6.2.1	Average Cache Hit Ratio by Using Four Topologies . . . . .	200
6.2.2	Average E2EDelay by Using Four Topologies . . . . .	205
6.2.3	Server Load Reduction by Using Four Topologies . . . . .	210
6.2.4	Total Reduction in the Network Footprint by Using Four Topologies	215
6.3	The Enhancement Percentage of the CPVP Mechanism . . . . .	219
6.4	Summary . . . . .	220
<b>CHAPTER SEVEN CONCLUSION AND FUTURE WORKS . . . . .</b>		<b>222</b>
7.1	Research Summary . . . . .	223
7.2	Research Contribution . . . . .	225
7.3	Research Limitations . . . . .	227
7.4	Future Works . . . . .	227
<b>REFERENCES . . . . .</b>		<b>229</b>

## List of Tables

Table 2.1	Coordination Caching Placement for VoD in NDN . . . . .	67
Table 2.2	Summary Comparison of Caching Placement Mechanisms . . . . .	70
Table 3.1	Comparison of Performance Evaluation Mechanisms . . . . .	84
Table 3.2	Simulation Parameter Settings . . . . .	97
Table 4.1	The Average Cache Hit Ratio for Seven Placement Mechanisms Using 1 GB to 1 TB and $\alpha = 0.65$ . . . . .	121
Table 4.2	The Average Cache Hit Ratio for Seven Placement Mechanisms Using 1 GB to 1 TB and $\alpha = 2.0$ . . . . .	123
Table 4.3	Average E2EDelay for Seven Placement Mechanisms Using 1 GB to 1 TB and $\alpha = 0.65$ . . . . .	128
Table 4.4	Average E2EDelay for Seven Placement Mechanisms Using 1 GB to 1 TB and $\alpha = 2.0$ . . . . .	129
Table 4.5	Server Load Reduction for Seven Placement Mechanisms Using 1 GB to 1 TB and $\alpha = 0.65$ . . . . .	133
Table 4.6	The Server Load Reduction for Seven Placement Mechanisms Using 1 GB to 1 TB and $\alpha = 2.0$ . . . . .	135
Table 4.7	Total Reduction in Network Footprint for Seven Placement Mechanisms Using 1 GB to 1 TB and $\alpha = 0.65$ . . . . .	140
Table 4.8	Total Reduction in Network Footprint for Seven Placement Mechanisms Using 1 GB to 1 TB and $\alpha = 2.0$ . . . . .	141
Table 5.1	Related Parameters in Model . . . . .	154
Table 5.2	Notation for Video Partitioning Scheme (VPS) . . . . .	184
Table 6.1	The Average Cache Hit Ratio for Four Placement Mechanisms Using 1 GB to 1 TB . . . . .	204
Table 6.2	Average E2EDelay for Four Placement Mechanisms Using 1GB to 1TB . . . . .	209
Table 6.3	Server Load Reduction for Four Placement Mechanisms Using 1GB to 1TB . . . . .	213

Table 6.4 Total Reduction in Network Footprint for Four Placement Mechanisms Using 1GB to 1TB . . . . . 218

Table 6.5 Summary of Enhancement Percentage of the CPVP Mechanism . . . 220



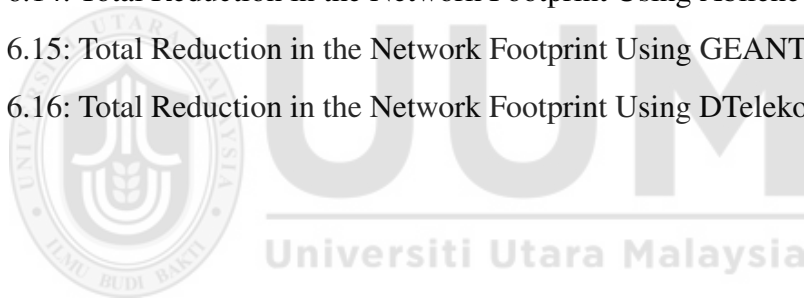
## List of Figures

Figure 1.1: The Statistic of Most Popular Video Type . . . . .	5
Figure 1.2: The Statistic of Video Game Market Revenue Worldwide . . . . .	5
Figure 1.3: The Architecture of Video Streaming Services over HTTP . . . . .	7
Figure 1.4: NDN Architecture . . . . .	8
Figure 1.5: The Scope of Study . . . . .	18
Figure 2.1: Classification of NDN . . . . .	26
Figure 2.2: Classification of Different Caching Mechanisms . . . . .	33
Figure 2.3: Caching Schemes . . . . .	40
Figure 2.4: The Operation of LCE . . . . .	48
Figure 2.5: The Operation of LCD, Prob, PProb . . . . .	51
Figure 2.6: The Operation of Centrality-Based Mechanism . . . . .	62
Figure 2.7: CBCP Evaluation . . . . .	64
Figure 2.8: Evaluation of a Cache Placement Mechanism Based on Entropy Weighting Method and TOPSIS . . . . .	65
Figure 3.1: The Questions-Steps-Objectives Diagram . . . . .	76
Figure 3.2: Conceptual Model of Coordination Placement of Video Partitioning Mechanism (CPVP) . . . . .	79
Figure 3.3: Main Steps for Validation and Verification . . . . .	83
Figure 3.4: Performance Evaluation Approaches . . . . .	85
Figure 3.5: Simulation Construction . . . . .	85
Figure 3.6: Structural Diagram of the ndnSIM2.0 Design Components . . . . .	88
Figure 3.7: The Source of Videos . . . . .	100
Figure 3.8: Example of DASH/SVC of BBB-I-360p Information . . . . .	101
Figure 4.1: The Operation of the LCD, Prob, and LCE Placement Mechanisms	113
Figure 4.2: The PProb Caching Mechanism Scenario . . . . .	114
Figure 4.3: Centrality-based Caching Mechanism Node Positioning . . . . .	114
Figure 4.4: Abilene Topology . . . . .	116

Figure 4.5: Average Cache Hit Ratio for Seven Placement Mechanisms Using 1 GB and $\alpha = 0.65$ . . . . .	117
Figure 4.6: The Average Cache Hit Ratio for Seven Placement Mechanisms Using 10 GB and $\alpha = 0.65$ . . . . .	118
Figure 4.7: Average Cache Hit Ratio for Seven Placement Mechanisms Using 100 GB and $\alpha = 0.65$ . . . . .	120
Figure 4.8: Average Cache Hit Ratio for Seven Placement Mechanisms Using 1 TB and $\alpha = 0.65$ . . . . .	121
Figure 4.9: The Average Cache Hit Ratio in Four Cache Size for Seven Placement Mechanisms with $\alpha = 0.65$ . . . . .	122
Figure 4.10: The Average Cache Hit Ratio for Seven Placement Mechanisms Using 1 GB to 1 TB and $\alpha = 2.0$ . . . . .	124
Figure 4.11: Average E2EDelay for Seven Placement Mechanisms Using 1 GB and $\alpha = 0.65$ . . . . .	125
Figure 4.12: Average E2EDelay for Seven Placement Mechanisms Using 10 GB and $\alpha = 0.65$ . . . . .	126
Figure 4.13: Average E2EDelay for Seven Placement Mechanisms Using 100 GB and $\alpha = 0.65$ . . . . .	127
Figure 4.14: Average E2EDelay for Seven Placement Mechanisms Using 1 TB and $\alpha = 0.65$ . . . . .	128
Figure 4.15: Average E2EDelay for Seven Placement Mechanisms Using 1 GB to 1 TB and $\alpha = 0.65$ . . . . .	129
Figure 4.16: Average E2EDelay for Seven Placement Mechanisms Using 1 GB to 1 TB and $\alpha = 2.0$ . . . . .	130
Figure 4.17: Server Load Reduction for Seven Placement Mechanisms Using 1 GB and $\alpha = 0.65$ . . . . .	131
Figure 4.18: Server Load Reduction for Seven Placement Mechanisms Using 10 GB and $\alpha = 0.65$ . . . . .	131
Figure 4.19: Server Load Reduction for Seven Placement Mechanisms Using 100 GB and $\alpha = 0.65$ . . . . .	132
Figure 4.20: Server Load Reduction for Seven Placement Mechanisms Using 1 TB and $\alpha = 0.65$ . . . . .	133

Figure 4.21: Server Load Reduction for Seven Placement Mechanisms Using 1 GB to 1 TB and $\alpha = 0.65$ . . . . .	134
Figure 4.22: Server Load Reduction for Seven Placement Mechanisms Using 1 GB to 1 TB and $\alpha = 2.0$ . . . . .	135
Figure 4.23: Total Reduction in Network Footprint for Seven Placement Mechanisms Using 1 GB and $\alpha = 0.65$ . . . . .	137
Figure 4.24: Total Reduction in Network Footprint for Seven Placement Mechanisms Using 10 GB and $\alpha = 0.65$ . . . . .	137
Figure 4.25: Total Reduction in Network Footprint for Seven Placement Mechanisms Using 100 GB and $\alpha = 0.65$ . . . . .	138
Figure 4.26: Total Reduction in Network Footprint for Seven Placement Mechanisms Using 1 TB and $\alpha = 0.65$ . . . . .	139
Figure 4.27: Total Reduction in Network Footprint for Seven Placement Mechanisms Using 1 GB to 1 TB and $\alpha = 0.65$ . . . . .	140
Figure 4.28: Total Reduction in Network Footprint for Seven Placement Mechanisms Using 1 GB to 1 TB and $\alpha = 2.0$ . . . . .	142
Figure 5.1: Tree Topology . . . . .	149
Figure 5.2: Coordinated Caching . . . . .	159
Figure 5.3: Non-coordinated Parallel Cache . . . . .	160
Figure 5.4: Non-coordinated Cache in Series . . . . .	160
Figure 5.5: The Cooperative Chunks in NDN Tables: (CIC) & (CDC) . . . . .	174
Figure 5.6: The Cooperative Pending Interest Table (CPIT) . . . . .	174
Figure 5.7: The Cooperative Content Store Table (CCS) . . . . .	175
Figure 5.8: Request for Video and Search in CS Along the Path to the Server in NDN . . . . .	177
Figure 5.9: Gathered the Partial Data Chunks . . . . .	178
Figure 5.10: Partition File on the Path . . . . .	182
Figure 5.11: Case1 When the Video Can Be Divided to Three Blocks . . . . .	191
Figure 5.12: Case2 When the Data Chunk Cannot Be Divided to Three Block Without Remainder . . . . .	193
Figure 5.13: The CPVP Implementation in ndnSIM2.0 . . . . .	197

Figure 6.1: Average Cache Hit Ratio Using Tree Topology . . . . .	201
Figure 6.2: Average Cache Hit Ratio Using Abilene Topology . . . . .	201
Figure 6.3: Average Cache Hit Ratio Using GEANT Topology . . . . .	202
Figure 6.4: Average Cache Hit Ratio Using DTelekom Topology . . . . .	203
Figure 6.5: Average E2EDelay Using Tree Topology . . . . .	206
Figure 6.6: Average E2EDelay Using Abilene Topology . . . . .	207
Figure 6.7: Average E2EDelay Using GEANT Topology . . . . .	207
Figure 6.8: Average E2EDelay Using DTelekom Topology . . . . .	208
Figure 6.9: Server Load Reduction Using Tree Topology . . . . .	211
Figure 6.10: Server Load Reduction Using Abilene Topology . . . . .	211
Figure 6.11: Server Load Reduction Using GEANT Topology . . . . .	212
Figure 6.12: Server Load Reduction Using DTelekom Topology . . . . .	212
Figure 6.13: Total Reduction in the Network Footprint Using Tree Topology . .	215
Figure 6.14: Total Reduction in the Network Footprint Using Abilene Topology	216
Figure 6.15: Total Reduction in the Network Footprint Using GEANT Topology	216
Figure 6.16: Total Reduction in the Network Footprint Using DTelekom Topology	217



## List of Abbreviations

ANDaNA	Anonymous Named Data Networking Application
AODV	Ad-hoc On-Demand Distance Vector
AWS	Amazon Web Services
BAS	Building Automation System
CBEWT	Cache Placement Mechanism Based on Entropy Weighting Method and TOPSIS
CBR	Constant Bit Rate
CBCP	Cache Placement Based on Compound Popularity Mechanism
CCFS	Controller-based Caching and Forwarding Scheme
CCN	Content-Centric Network
CCS	Cooperative Content Store Table
CDC	Cooperative Data Chunk Table
CDN	Content Delivery Network
Centrality	Centrality-based mechanism
CIC	Cooperative Interest Chunk Table
CL4M	Cache Less for More
COBRA	COntentdriven Bloom filter-based intra-domain Routing Algorithm
COMET	COntent Mediator architecture for content-aware nET-works
CONET	COntent centric inter-NETworking architecture
CoRC	Coordinated Routing and Caching
CPIT	Cooperative Pending Interest Table
CPM	Content Placement Mechanism
CPVP	Coordination Placement by Video Partitioning Mechanism
CR	Content Router
Cross	Hybrid caching
CS	Content Store
DASH	Dynamic Adaptive Streaming over HTTP
DONA	Data Oriented Network Architecture
E2EDelay	End-to-End Delay
EgoNet	Egocentric Network study software

FIB	Forwarding Information Base
GB	Giga Byte
HTTP	Hypertext Transfer Protocol
HVAC	Heating, Ventilation, and Air Conditioning
ICN	Information-Centric Network
INFORM	INterest FORwarding Mechanism for Information-Centric Networking
IoE	Internet of Everything
IoT	Internet of Thing
IP	Internet Protocol
ISP	Internet Service Provider
Kb	Kilo bit
LCD	Leave Copy Down
LCE	Leave Copy Everywhere
LFU	Least Frequently Used
LRU	Least Recently Used
MZipf	Maldelbrot-Zipf
Mb	Mega bit
MPD	Media Presentation Description
MPEG-DASH	MPEG-Dynamic Adaptive Streaming
NDN	Named Data Networking
ndn-cxx	NDN C++ library with eXperimental eXtentions
ndnSIM2.0	NDN Simulator version 2.0
NetInf	Network Information
NFD	NDN Forwarding Daemon
non-CBR	Non-Constant Bit Rate
NRENs	Europe's Research & Education Networks
NS3	Network Simulation version 3
NSCC	Not So Cooperative Cache
OTT	Over-The-Top (OTT) media service
PARC	Palo Alto Research Center
PIT	Pending Interest Table

PProb	Path Probabilistic cache
Prob	Probabilistic cache
ProbCache	Content-Aware Caching
PSIRP	Publish and Subscribe Internet Routing Paradigm
PT	Popularity Table
PURSUIT	PUBlish Routing SUBscribe Internet Technology
QoE	Quality of Experience
QoS	Quality of Service
QSOD	Questions-Steps-Objectives-Diagram
Rand	Random choice caching
RSA	Rivest Shamir Adleman
RTT	Round Trip Time
SAIL	Scalable and Adaptive Internet Solutions
SD	Standard-Definition
SSD	Smart System Development
SSH	Secure SHell
STEM	Science, Technology, Engineering, and Mathematics
SVC-DASH	Scalable Video Coding-Dynamic Adaptive Streaming over HTTP
SVoD	Subscriptions Video on Demand
TB	Tera Byte
TCP	Transmission Control Protocol
TelNet	Teletype Network
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution
TV	TeleVision
UCLA	University of California in Los Angeles
UHD	Ultra-High Definition
VoD	Video on Demand
VoIP	Voice over Internet Protocol

# CHAPTER ONE

## INTRODUCTION

Over the past decades, the Internet has become one of the most popular tools that are utilised in our everyday lives. With its development being based on Transmission Control Protocol/Internet Protocol (TCP/IP), the Internet has flourished dramatically. This progress is actualised not just because the Internet has many applications, but also because of the different media types that have access to it. However, with this expansive growth and diversification, the network exposes several significant challenges, such as delay, data redundancy, and unexpected interruption. Hence, the focus is on developing and growing future network research [1, 2]. Furthermore, coordinated caching within Named Data Networking (NDN) for Video-on-Demand (VoD) is crucial for enhancing performance, scalability, and user experience. Coordinated caching is a strategic and cooperative approach where caches at different network nodes work together, rather than operating independently, to manage what content is stored, replicated, or evicted.

The initial target of the Internet was to share data between two point-to-point endpoints and offer channels of communication between them. This concept follows the traditional telecommunication network idea. In this process, the function of the identifier and locator of the data holders is handled by the IP address, which is the most essential element for the transmission of packets. For consumers to access data, they need to be initially informed about the location of the data and also be provided with the IP address of the holder. However, many consumers are only concerned about what they can subscribe to and less about where they can obtain the content [3, 4].

To enhance serving the consumer, the development of future networks is needed, such as NDN, which is a new architecture developed by the Palo Alto Research Center

(PARC) [5, 6, 7]. NDN advocates an information-centric communication model rather than a host-centric one on the Internet. In-network caching, as an essential component of the information-centric communication model, enables routers to cache content replicas and respond to subsequent content requests with matched content. Not only does NDN reduce content distribution costs and optimise network performance, but it also significantly improves the end-user service experience [6, 8, 9]. However, caching in NDN is more dynamic, open, and flexible. These features are due to their application-agnostic nature, ubiquitousness, and network awareness. Certain advantages of NDN include the data copy being closest to the consumers, reducing traffic on the original server, caching content everywhere, and enabling sharing from any network node.

Although the NDN is still in its early foundation stage, details, including the cache issue, should be considered. There is still more work that needs to be done. As a result of all network nodes being able to cache the content, caching tends to become more complicated due to deciding if the content will be cached and in which nodes, or if the node will cache the whole content, or if multiple nodes will cooperate to cache the content. Hence, the need to introduce approaches for improving caching placement performance in NDN arises.

While NDN addresses broad networking challenges, specific applications, such as VoD services, bring their considerations instead of broadcast television. However, a VoD service grants the user more authority and responsibility by allowing them to select from a wide range of shows, necessitating active decision-making. One issue that can arise is that VoD customers who have trouble making choices while browsing for content to view may end up sticking with their go-to selections and eventually unsubscribing [10, 11]. In a VoD service, this problem could be more critical since there is a need to see the video directly without any delay in delivery time, bearing in

mind that the video requires a large cache size. Thus, improving the caching placement mechanism is an increasing challenge, raising the question: Is caching the video by cooperating between multiple nodes best due to the limited cache size and large video size?

Addressing the aforementioned challenges is critical, particularly in the context of NDN caching for VoD services. This thesis proposes a new caching placement mechanism to reduce the average End-to-End Delay (E2EDelay) performance in NDN during VoD streaming. The reduction in delay is critical for consumers to get quality service by receiving video directly over the Internet. As a result, this chapter presents a research overview that covers the online VoD architecture, a brief introduction to NDN and its placement mechanisms, and the study of historical context. This chapter also outlines the research problem, objectives, and questions. The scope and significance of the research are then illustrated briefly. Finally, this chapter ends with the outline of the thesis.

## **1.1 Background of the Study**

Internet video services mean using a video on Internet architectures that are already popular, like the Hypertext Transfer Protocol (HTTP) and TCP architectures [12, 13]. These architectures are generally available on Netflix, Vudu, and YouTube due to their ease of streaming video to Content Distribution Networks (CDN) [9, 10, 14]. The VoD platforms employed Over-The-Top (OTT) streaming. OTT streaming refers to the arrival of content over the Internet instead of via satellite subscription or classic cable television [15]. Psaras et al. [16] pointed out that flooding video over the Internet via HTTP offers numerous benefits, including uniformity across the CDN in mobile video streaming services. CDN is also accessible all over the world. Finally, because of the service's commercialisation and competitive pricing, CDN remains cost-effective.

These benefits have caused enormous growth in transmitting high-quality movies and streaming TeleVision (TV) at affordable prices, which benefits the viewer [17, 18].

In addition, Internet video is generally characterised by traffic problems, unlike conventional video, which requires it to remain fully present locally before playback. Specific research indicates that online videos are encoded at varying speeds, spanning from standard definition to high definition. The standard definition of video over the Internet is 235 Kb/second, while the high definition is 5 Mb/second [19, 20]. The video store has to hold many files on the device, which creates extra work for the cache. The high storage requirements of Internet video transmission also lead to error recording [21, 22]. The research on Internet traffic showed that most of the activities carried out on the Internet involve viewing, downloading, and uploading videos.

Furthermore, the number of online videos has increased in recent years. In 2023, Statista estimates that over three billion people will view, download, or stream video at least once a month. Meanwhile, 92% of all Internet users globally viewed online video in 2024 [23]. Figure 1.1 represents the ratio of each type of video. Thus, there is an increasing load on content providers, which implies watching or downloading movies from private users over the Internet to advertise, learn, or engage in government activities [6, 7].

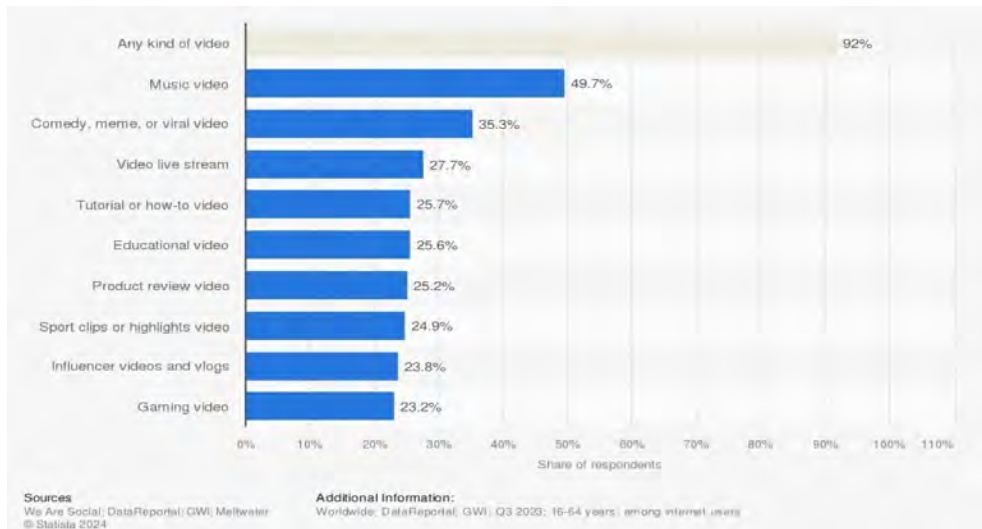


Figure 1.1. The Statistic of Most Popular Video Type

Even though just 23.2% of all videos are gaming-related [23], global revenue in the video market's "Games" section is expected to increase between 2024 and 2029 [24], reaching 211.4 billion dollars, or a 46.43% increase. The indicator is expected to set a new high in 2029, reaching 666.69 billion U.S. dollars after growing for seven years. Figure 1.2 illustrates this data. Thus, there is an increasing load on content providers, which implies watching or downloading movies from private users over the Internet to advertise, learn, or engage in government activities [6, 7].

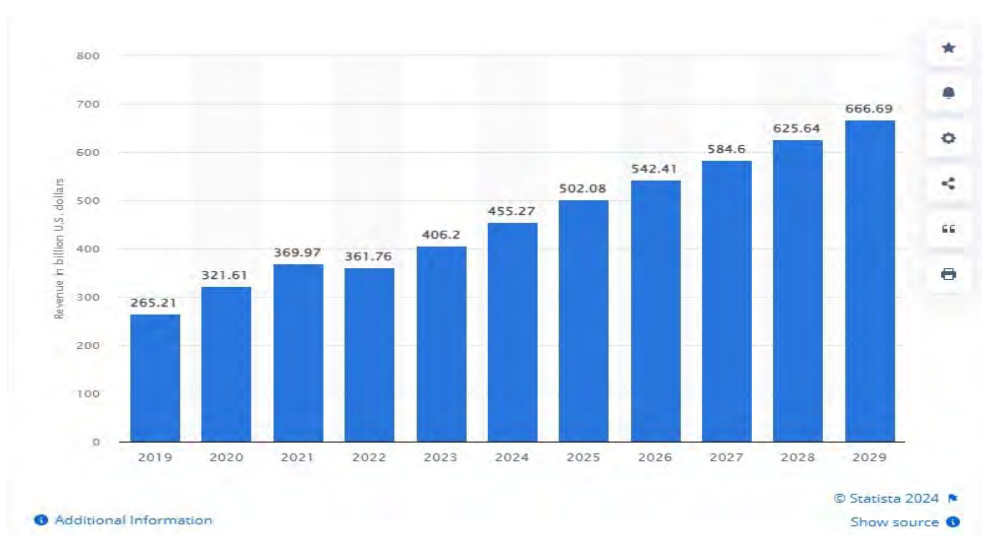


Figure 1.2. The Statistic of Video Game Market Revenue Worldwide

To ensure efficient content delivery, especially with the rising demand for online videos, the NDN must evolve to enhance caching by placing the content in the right node [9, 13, 12]. This is critical because the increasing accessibility of online videos induces viewers to insist on their desired video content. Such a surge in demand leads to workload overload. Consequently, caches must provide content at a very precise time, which can lead to delays in receiving data packets. Therefore, the primary issue lies in how caching can reduce the high delivery time of VoD traffic, which is a result of various user requests for online videos.

## **1.2 Video on Demand in NDN**

In the past decade, researchers have targeted the NDN architecture for the future Internet [25, 16, 26]. Therefore, Cheriton and Gritter's research [27] validated the initial proposal for this concept. Current Internet services extensively explore and employ the CDN as a substitute method [22]. However, due to the lack of coordination among multiple CDN service providers [20], the CDN architecture has limitations on where to place its contents and services to support consumers.

This section covers the overall NDN and CDN architecture. Figure 1.3 illustrates the most common video-streaming architectural facilities. A typical video service supports several platforms, including gaming consoles, TVs, and browser plug-ins. Authentication and streaming are the two stages of a video session. When a client requests a video, the service's authenticity consumes a CDN proportionate to the video. The video streaming provider notifies the client of the video streaming proportions offered and issues a Token for each proportion. The customer then selects a particular video proportion and demands video at that proportion by providing a token as a validation certificate to the specified CDN [28, 29].

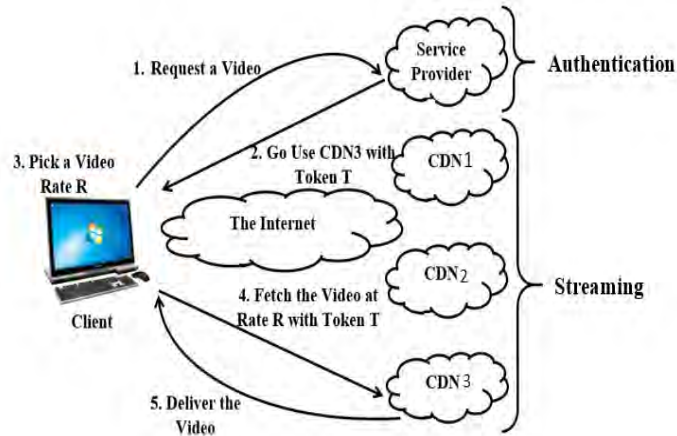


Figure 1.3. The Architecture of Video Streaming Services over HTTP [28, 30]

However, the NDN represents the Internet's future, expanding upon the concept of the CDN. The NDN anticipates a hierarchical design for names. For example, the University of California, Los Angeles (UCLA) [30] could establish a video with a name similar to /ucla/videos/demo.mpg, with the '/' character determining the name elements. NDN construction contains two main association units: a Data packet and an Interest packet. These association units carry hierarchically designed names [30, 28]. The system transmits an Interest packet when a user requests Data. A Data packet can be used to complete an Interest packet; as a result, the name conveying Interest is the same as the video's title. Additionally, when many Data packets potentially fulfill the Interest, an Interest packet might send a particular field to establish priorities [5, 25].

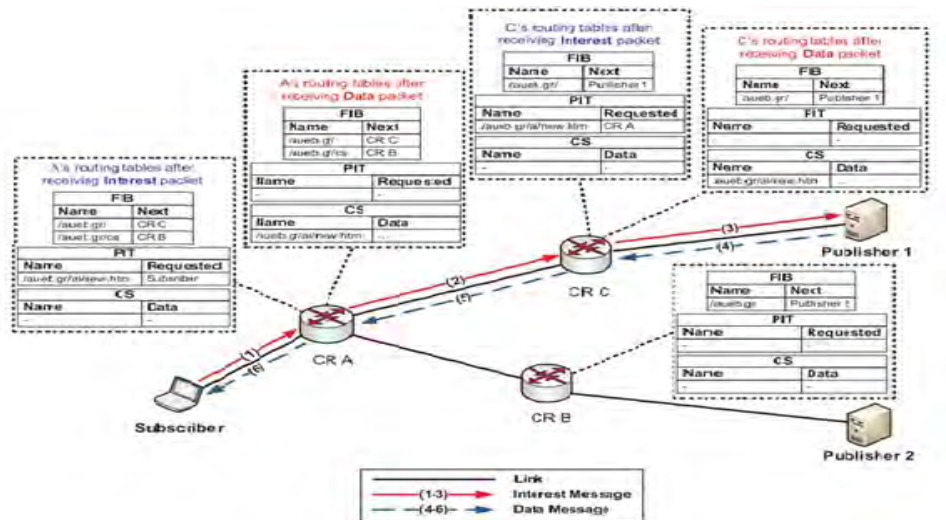


Figure 1.4. NDN Architecture [31, 32]

In addition, Figure 1.4 illustrates the flow of the NDN architecture's operations. The user transmits Interest on /aueb.gr/ai/new.htm as arrows 1-3 show. When a packet becomes interesting, the CR learns the name of the Interest packet and searches for it in the Content Store (CS). If an Interest packet is delivered back over the input interface to a Data packet and matches the prefix in CS, the Interest packet is cancelled [31, 32]. The router transmits a longer routing prefix to its Forwarding Information Base (FIB) if the prefix does not match the interested packet to ascertain its destination number. When a record is established in the FIB, the router registers the arrival interface of the Interest packet in the Pending Interest Table (PIT). The FIB specifies that this packet goes to the CR. On the other hand, if the prefix is matched at a producer node or a CS, then the Interest packet is cancelled, and the Data is received in a packet of Data [33, 32].

Additionally, the CR does a more extensive prefix matching with its PIT to discover an input relevant to the Data packet after recording the associated prefix of the retrieved data message in its content store. When this PIT record lists several interfaces, a PIT duplicates the Data packet and achieves multicast distribution. Finally, as shown

in arrows 4-6, the content router transmits the data packet to these interfaces while deleting the record from the PIT. The router eliminates the Data packet as a copy if the matching is not recorded in the PIT [32, 34]. At the same time, the NDN's broadcasts concentrate on the receiver. When a user submits an Interest packet, a Data packet can be acquired.

Moreover, the routers log the transmission of an Interest packet in the PIT [31, 33], identify the channel from which the packet originates, and employ a forwarding scheme to decide where to forward the interest. The cache placement mechanisms determine whether to cache the data and which routers to store it on. Thus, one can return data packets by retracing the reverse path to their original request location. One or more routers use a cache placement mechanism to cache whole data packets during the reverse route. This procedure creates a summary multicast tree for each requested data item using the exact data provided by all consumers. Furthermore, this requested content may consist of multiple data packets, particularly if it is a video [31, 35, 36].

While NDN guarantees superior performance for consumers by storing video data in closer proximity than CDN, it necessitates a significant increase in operations between requesting the desired packet and receiving it. Due to the large size of videos and the limited cache size of each router, these numerous operations cause server load and average E2EDelay, especially when the requests are video-based. Thus, an important question arises. How can a large video cache fit into a limited CS? What measures can we implement to improve the cache placement mechanism? Researchers [37, 38, 39] emphasised the importance of data placement coordination for optimal system performance. Chen et al. [40] demonstrate that cooperative caching can more efficiently utilise limited storage and satisfy different user needs for video.

However, a streaming application experiences a reduction in quality preferences when

it receives data from a longer path with poor video source performance. Regardless, if a consumer requests low-bitrate video, they can receive high-quality content by coordinating the retrieval from nearby caches. Therefore, to shorten this long path and reduce the overall network footprint, there is a need for a new cooperative cache mechanism. The reduction in the network's footprint metrics is an important aspect of network management, as it helps with resource optimisation, performance enhancement, and operational cost reduction by minimising the impact of data traffic. This metric involves a combination of techniques aimed at making data transmission more efficient and sustainable. Furthermore, unstable network conditions, such as intermittent cache hits and misses, contribute to instability dynamics [41, 42, 43]. Therefore, the partitioning mechanism is necessary to effectively increment the cache hit rate [44, 45]. This increases the motivation to design a new coordinated cache partitioning algorithm for effective performance.

### **1.3 Research Motivation**

NDN aims to offer solutions to certain challenges and problems on the Internet nowadays, such as content delivery, workload, latency, and security. This network's nodes store data for future requests. However, numerous NDN research challenges remain unresolved, including caching and naming challenges in routing [6, 46]. The challenges of caching in NDN remain a key area of future research.

Note that NDN is not an infallible means of addressing Internet problems; many technical and research problems still need efficient answers. For this reason, the large difference between scarce routers' resources and the enormous space of content names is still one of the major challenges for NDN deployment on a large practical scale. Content addressing, naming, and universal caching are additional challenges that NDN introduces to routing. Among these, universal caching, also known as in-

network caching, has garnered significant research interest in recent years. This is due to the longstanding recognition of caching as an effective solution to the scalability issue in large-scale systems. Another reason why universal caching has gained so much attention is because it directly influences content distribution efficiency, which simultaneously impacts the end user's experience in many Internet applications. Therefore, in NDN architecture, caching is considered an indispensable component [47, 48]. In addition, the challenge of ensuring that the publish/subscribe paradigm is both scalable and efficient for video transmission in normal and disaster scenarios [49, 50, 51].

A few years ago, the rise of social media networks (like Facebook, Instagram, and Twitter), entertainment applications, and multimedia content providers (like Netflix, YouTube, VoD, etc.) led to a large demand for video traffic on wireless mobile networks [52]. With this high demand, there is an expectation that there will be an increase in traffic. However, the increasing use of multi-device devices and connections is significantly impacting traffic patterns, particularly on video devices. The device's video effects on traffic are more pronounced because of the introduction of Ultra-High Definition (UHD), or 4K, video streaming. With this high demand, there is an expectation that there will be an increase in traffic. However, the increasing use of multi-device devices and connections is significantly impacting traffic patterns, particularly on video devices. The device's video effects on traffic are more pronounced because of the introduction of UHD, or 4K, video streaming. This technology has such an effect because the bit rate for 4K video at about 15 to 18 Mbps is more than double the HD video bit rate and nine times more than the Standard-Definition (SD) video bit rate. By 2023, two-thirds of the installed flat-panel TV sets will be UHD, up from 33% in 2018 to 66% [53]. These statistics form the basis for a large motivation to investigate how to operate efficiently in the presence of such traffic.

By 2024, all types of multimedia traffic, including TV and VoD, would account for over 92% of all consumer traffic worldwide [23]. The volume of Internet traffic rises by a factor of 1.7 per year [54]. For instance, according to a report, the percentage of delayed videos and video quality are rising simultaneously. The statistics display 45.7% of 720-pixel (p) videos stalling and 10.5% of 240-pixel videos likewise experiencing a stall [55, 56].

The most challenging issue in ensuring a seamless online video viewing experience is the large size of video files and the need to address stalling within a few milliseconds. Therefore, the question is how to satisfy these requests for video with video content delivered on time and with high encoding quality. This will increase the users' satisfaction and increase the expected revenue. Complex network decisions are challenging and require careful consideration to provide each user with the appropriate data rate based on their unique channel circumstances. Transmitting high-quality video files increases network demand, leading to occasional maintenance expenditures [57, 58]. This provides a further motive, since, for obvious economic reasons, it is critical to predict how most Internet VoD systems will function in the future [59, 60]. This is because a delay in receiving video can reduce video quality and potentially cause a loss of data packets [61, 62]. Thus, all these factors result in significant financial losses, particularly when VoD systems initiate requests.

The primary method for addressing prolonged memory access latency in computing systems [28, 13]. Additionally, having multiple degrees of customisation for the resource's components may result in extra design costs and performance requirements [63]. Earlier research indicated the necessity for efficient, scalable cache coherence protocols that may function in NDN caches to execute multiple copies of a cache at various levels of the cache hierarchy [55, 64, 63].

The challenge of ensuring high-quality video and data, reduced delay, increased speed, and improved network utilisation is significant. However, cache performance suffers as video consumers dynamically choose content encoded at different bitrates. Researchers have found that partitioning the NDN cache can be useful along a forwarding path into separate sections to solve a bitrate oscillation [42, 41]. In addition, it is crucial to safeguard various aspects of the cache's capacity, such as improving the footprint, optimising cache placement, and reducing bitrate oscillation. Coordinating and allocating cache space to specific routers within a suitable range of hops from consumers presents additional challenges. These include stabilising bandwidth fluctuations, reducing bitrate oscillation, and subsequently reducing the average E2EDelay.

Despite numerous studies and improvements to optimise the Internet for video transmission applications, the majority of research relied solely on network performance measures. Regardless of the NDN advantages, it creates various challenges, such as designing a new data caching mechanism [65, 66]. Another driving force behind this study is the lack of research on NDN and VoD's role in user-perceived quality video performance delays.

Therefore, this thesis examines the effectiveness of various cache placement mechanisms in optimising their performance. Subsequently, it formulates, evaluates, and projects a novel coordinated caching placement mechanism for VoD applications.

#### **1.4 Problem Statement**

VoD services have become a major driver of Internet traffic, requiring efficient caching mechanisms to meet the growing demand [67, 68]. NDN, with its in-network caching capabilities, provides a promising framework for addressing these [17, 69, 70]. However, the inherent complexity of VoD workloads, including high data volumes

and simultaneous requests, exacerbates inefficiencies in traditional caching approaches [71]. NDN facilitates more effective sender-receiver communication during online video requests. Furthermore, caching online video requests in NDN has proven to be more efficient than in traditional Internet architectures, offering a significant advantage [72, 33, 73, 74]. Content redundancy and inefficient routing paths make this stress even worse, leading to long the average E2EDelays, heavier server loads, and even sudden network interruptions [75, 76, 77, 78, 79, 80]. This study attempts to improve the overall caching mechanism for VoD services in NDN, while VoD is one of the application services, emphasising the importance and timeliness of the research.

Existing caching mechanisms, caching placement mechanisms can be classified according to the coordination concept, such as non-coordinated, implicitly coordinated, and explicitly coordinated. Non-coordinated mechanisms, such as Leave Copy Everywhere (LCE), have high redundancy and inefficient use of network resources. This increases because each router independently operates the canonical caching mechanism [81, 32]. However, implicit cache coordinated schemes like Leave Copy Downstair (LCD) and Probabilistic (Prob) allow nodes to exchange information before determining storage locations [82, 16, 83]. LCD fails to optimise caching placement dynamically across the network, while Prob helps process greater cache volumes. Prob effectively reduces network traffic redundancy and minimises delay time compared to current placement mechanisms [84]. However, Prob still needs to investigate the best location for the cache due to the large size of VoD [85]. Explicit cache-coordinated schemes, such as partitioning videos with a limited cache size on multiple routers, can improve memory utilisation by inserting required information into requested content [86, 38]. These shortcomings highlight the need for explicit coordinated caching mechanisms capable of improving resource utilisation and reducing latency.

Previous studies have investigated coordinated caching strategies in NDN for VoD services, including content popularity-based approaches and fair cache distribution mechanisms. For example, research by Li and Wang (2012) proposed a caching scheme leveraging content popularity to reduce inter-ISP traffic [87], while Jiang et al. (2017) explored equitable cache allocation among users [88]. These studies demonstrate the potential of coordinated caching to enhance network performance. However, existing approaches often overlook the benefits of integrating dynamic video partitioning, strategic caching placement, and aggregation mechanisms into a unified framework. Additionally, limitations in handling VoD-specific challenges, such as redundancy reduction and latency minimisation, remain inadequately addressed.

To address these challenges, a coordinated approach to caching is essential by using hypergraphs, which can effectively model cooperative behaviour. However, each hyperedge in a hypergraph can represent a group of nodes that collaborate to cache a particular piece of content, such as VoD. Hypergraph takes into account the connections between nodes and establishes a relationship between the edge nodes and the midpoint nodes [89]. This new coordinated approach dynamically partitions video content, strategically placing caches, and aggregates content from multiple nodes.

On the other hand, higher latency makes users less satisfied with QoE and QoS [90, 91, 92, 93] because QoS-aware multimedia apps like VoD need to be sent with less delay [94, 95, 96]. This less delay avoids dropped data and keeps the video's bitrate as required as possible, which goes over the network's bandwidth limit and prevents the stop playback [97]. QoE refers to a user's overall experience when using a service or product, while QoS measures the performance of a service or network. Therefore, there are significant challenges to enhancing caching placement mechanisms to gain efficient network performance. However, the potential of cooperative caching is significant. By caching the video at the nearest nodes to the consumer, it is possible to achieve

this improvement by lowering data retrieval latency and reducing hop distance. In addition, partitioning the video between the cooperating nodes can minimise server load, increase the cache hit ratio, and decrease the aggregate network traffic load. This potential of the proposed mechanism to significantly improve network performance should inspire hope for the future of NDN and VoD services [98, 92, 99]. Therefore, because the source of coordinated caching in NDN is closer to the server path, caching the content using a proper placement mechanism does not require additional forwarding to the content source. VoD has revolutionised education by empowering students to participate actively in their education.

This study focuses on the development of a coordinated caching placement mechanism designed to improve VoD performance in NDN. By addressing the identified gaps, this research aims to optimise caching efficiency, reduce latency, and enhance user experience in high-demand VoD scenarios.

### **1.5 Research Questions**

This study examines the functional requirements, which are the specific skills and features a system needs to meet users' needs. These requirements will then be used to develop a new coordinated mechanism based on a set of rules and steps for reaching a clear goal of caching the desired video content in NDN in high-demand VoD environments. This will improve the user experience and make the network more efficient. Whereas user experience refers to how smoothly and satisfyingly a viewer can access and watch video content through an NDN-based network architecture.

1. What are the functional requirements to improve the NDN coordinated caching placement performance regarding the average E2EDelay in high-demand VoD environments?

2. How can functional requirements be employed to propose a coordinated mechanism to cache the interested video content in NDN in high-demand VoD environments, thereby improving the user experience and network efficiency?
  - a) How can fetching the interested video content increase the total network footprint reduction?
  - b) What is the caching decision that can help reduce the average E2EDelay?
  - c) How can partitioning video help increase the average hit ratio and the server load reduction?
3. What is the effect of the proposed mechanism in terms of the total reduction in network footprint, the average E2EDelay, the average hit ratio, and server load reduction?

## 1.6 Research Objectives

This research aims to create a new cooperative NDN caching placement mechanism for VoD. The new mechanism is intended to be applied to VoD to minimize the average E2EDelay. The following research goals will be the result of this research:

1. To empirically evaluate existing NDN caching placement mechanisms (LCD, Prob, LCE, Cross, Pprob, Rand, and Centrality) to identify specific functional requirements, such as redundancy reduction and performance optimisation, for coordinated caching mechanisms tailored to VoD workloads.
2. To design a novel caching mechanism called the Coordination Caching Placement by Video Partitioning (CPVP) to address inefficiencies in caching and improve network performance metrics such as the average E2EDelay, cache hit ratio, and server load. The CPVP integrates coordinated strategies, including video aggregation, caching decisions, and dynamic partitioning, tailored for VoD workloads in NDN. The CPVP mechanism incorporates findings from the

empirical analysis of existing mechanisms, addressing identified gaps such as limited coordination and inefficient caching placement in high-demand VoD scenarios.

3. To implement and evaluate the performance of the proposed NDN explicit coordinated caching placement mechanism (CPVP) and compare it with non-cooperated placement (LCE), and implicit coordinated placement (LCD, and Prob) in a simulated network environment for VoD.

### 1.7 Research Scope

This study focuses on the design and implementation of a new caching placement mechanism to enhance the performance of the Named Data Network during VoD workloads. The research scope is shown in Figure 1.5. As shown in the figure, this study gives a general overview of NDN in ICN. The conventional architecture of NDN stores online video consumers' requests through Internet VoD services. Although recurrent requests by the users often delay the performance of the videos during VoD workloads [17, 100].

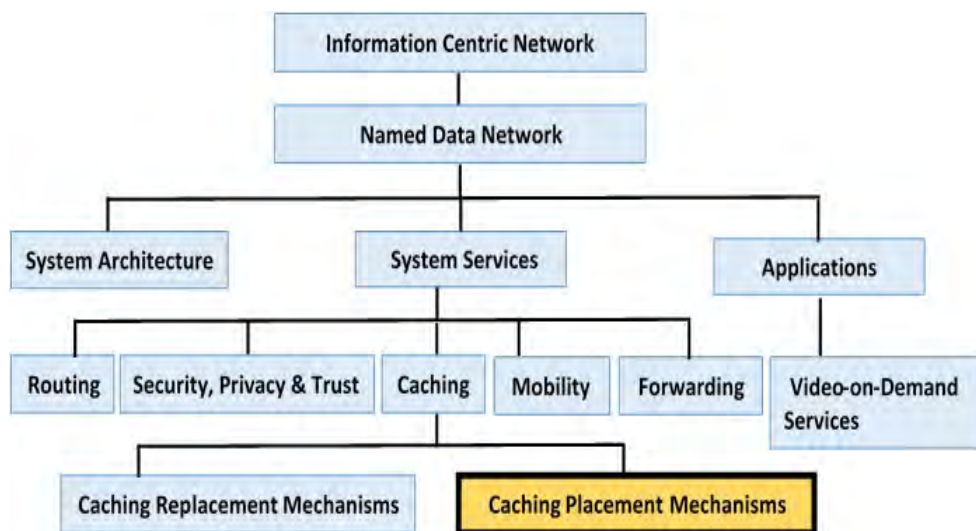


Figure 1.5. The Scope of Study

The primary focus of the work is on caching placement mechanisms; caching

replacement mechanisms are not included. Chapter 2 will thoroughly explain and analyse the nine cache placement mechanisms with VoD. These mechanisms are LCD, Prob, LCE, hybrid caching (Cross), Path Probabilistic cache (Pprob), Random choice caching (Rand), Centrality, Cache Placement Based on Compound Popularity Mechanism (CBCP), and Cache Placement Mechanism Based on Entropy Weighting Method and TOPSIS (CBEWT). This further enhances the NDN caching placement mechanism to reduce the delay time. First, this study will evaluate the available cache placement mechanisms regarding delay performance during the VoD workload. Then, go further in designing and implementing a new caching placement mechanism in NDN during VoD and evaluate it.

### **1.8 Significance of the Research**

This study is significant in its aim to significantly improve the NDN architecture by thoroughly reviewing current placement mechanisms, especially focusing on how they work within VoD services. The new Coordinated CPVP was designed specifically for VoD services in NDN, and its potential impact on Internet architecture and Internet of Things (IoT) networks cannot be overstated. The proposed coordinated placement mechanism will enhance resource delivery by reducing delay performance, leading to more trust in future VoD services. This study will be a crucial aid in actualising the provision of Internet architecture and improving the VoD for the Internet of Everything (IoE) and the IoT. On the other hand, IoE and IoT involve connecting physical devices and applications. However, IoE takes this further by utilising these connected devices as part of a larger, more intelligent system, such as cameras as physical devices and VoD as the application. The researchers Naeem et al. [101] foresee NDN as a promising design to safeguard IoT networks, given its rapid expansion and potential.

The CPVP enhances VoD services, making them more beneficial in various fields

such as the arts, government, and sociology. VoD preserves cultural performances, facilitates public information sharing, and allows for the documentation and analysis of social behaviours. Schools can save on bandwidth expenditures by acquiring telecommunication tools and computer systems. VoD has academic, social, and moral implications, including curriculum development, search engine designs, and school policies on information retrieval. Teachers using the Internet as an educational tool also benefit from VoD. Classroom films foster interaction with the material, benefiting cognitive growth. In real-time, VoD allows students to take an active role in their education, making the learning process more engaging and active.

In the rapidly evolving domain of large-scale content-centric networks, VoD services grapple with significant challenges. These pertain to VoD performance delays, burgeoning video traffic, and content distribution constraints. A key issue, for instance, is the burgeoning demand for storage due to the ever-increasing user appetite for online films. This research seeks to address these challenges head-on. The CPVP mechanism optimises storage utilisation, reduces VoD access delays, and enhances overall service quality. Such improvements are anticipated to benefit users and providers of online videos as the proposed mechanism becomes a standard in NDN, which is expected to spearhead advancements benefiting the entire NDN research community.

Therefore, this study will emphasise solving the challenges facing online video traffic that affect users and online video host providers. The issue must be resolved since the response time to requests for VoD traffic is shortened because CPVP caches the video at the coordinated edge routers. The CPVP mechanism helps minimise average E2EDelay and increases the total network footprint reduction, the average hit ratio, and the server load reduction. Importantly, CPVP will create direct income for Internet providers and generate indirect revenue for online video hosts. This potential for revenue generation should motivate and inspire all involved. Most significantly,

CPVP will enhance QoS and QoE, ensuring a better user experience and network performance.

Developing the coordinated cache placement mechanism improves the NDN architecture by significantly addressing the current delay time issues and unnecessary placement. The concept of NDN coordinated placement caching, especially when using CPVP, has the potential to make a profound impact. As CPVP becomes the standard, the entire NDN research community can look forward to reaping its benefits.

## **1.9 Research Contributions**

The primary motivation is to address the challenges associated with coordinated cache placement for VoD in NDN. This study's main contribution, creating a new Coordinated Cache Placement by Partitioning mechanism, CPVP, is a significant advancement that enhances NDN's functionality. The functional requirements are discovered by experimental evaluation of the existing cache placement mechanisms. This study will also be an important aid in actualising the provision of Internet architecture and improving the VoD for the IoE and the IoT.

The CPVP enhances VoD services, empowering students to take an active role in their education and making the learning process more engaging and active. CPVP requires three schemes: VAS, VCDS, and VPS. VAS helps aggregate the interested video, reducing path and data redundancy to reduce the network footprint. VCDS decides to either cache in the edge location or not cache to reduce the average E2EDelay. VPS works on partitioning video into three routers to increase the average hit ratio and reduce server load. Thus, the CPVP mechanism optimises storage utilisation, reduces VoD access delays, and enhances overall service quality.

Such improvements are anticipated to benefit users and providers of online videos as

the proposed mechanism becomes a standard in NDN. This standardisation is expected to spearhead advancements that will benefit the entire NDN research community, offering a promising future. It is also important to consider the contribution of this academic study to scientific knowledge in this area.

### **1.10 Thesis Outline**

This thesis is arranged into seven chapters. These chapters are organised as follows:

**Chapter One** provides an introduction to the topic, together with the background of the study, problem statement, research questions, and objectives. The chapter also includes the research scope, significance, and thesis outline.

**Chapter Two** reviews existing studies in the literature dealing with NDN and in-network caching, then focuses on analysing many caching placement mechanisms with VoD services.

**Chapter Three** emphasises the methodology approach to fulfil the research objectives and the simulation set in this research work.

**Chapter Four** focuses on the empirical evaluation of seven caching placement mechanisms in NDN during VoD to identify the features of these mechanisms. Thus, the first objective will be achieved.

**Chapter Five** presents the network model and analyses the problem of the caching placement mechanism, then analyses the coordination and non-coordination

placement mechanisms to design the new coordinated video partitioning mechanism called CPVP, which has three schemes: VAS, VCDS, and VPS. Thus, this chapter aims to satisfy the second objective along with its verification and validation.

**Chapter Six** evaluates the CPVP performance and compares it with the LCE, LCD, and Prob caching placement mechanisms by simulations. The performance metrics for evaluation are the average E2EDelay, average cache hit ratio, total reduction in the network footprint, and the server load reduction during VoD traffic. This chapter achieves to satisfy the third objective.

**Chapter Seven** presents the research conclusion and a summary of the contributions and limitations, in addition to offering more future studies based on the results of the current study and the research reduction in the network footprint and the server load reduction during VoD traffic. This chapter achieves the third objective, issues related to NDN fundamentals.

## **CHAPTER TWO**

### **LITERATURE REVIEW**

The Internet is transforming significantly from the traditional host-to-host communication model to a content distribution platform. This shift is driven by the increasing volume of Internet traffic that is now primarily hosted and location-independent. This is a departure from the traditional applications that rely on machine-to-machine or point-to-point connections, such as Teletype Network (TelNet), Secure Shell (SSH), and Voice over Internet Protocol (VoIP). The implications of this shift are profound, as it not only changes the way of using the Internet but also necessitates a rethinking of Internet architecture and the development of new networking paradigms like Information-Centric Networking (ICN) [102, 103].

This Internet traffic embodies examples that include the transfer of data in bulk, the web, and video distribution, not forgetting the current advances and trends on social networking platforms. To accommodate the importance of introducing a content-oriented Internet, recent studies conducted by researchers have put considerable effort into reconsidering Internet architecture, bearing in mind that the dominant usage pattern is user-to-content traffic. ICN, being a unique networking paradigm, prioritises content at the forefront of any communication initiated between random dual users of the system without paying attention to concepts of host- and location-identifiers [104, 105].

In ICN, content is named explicitly and uniquely, enabling users to request content using these identifiers rather than the traditional node identifiers used in Internet Protocol (IP) addresses [106]. ICNs enable many nodes, especially those close to the end users, making communication and storage provision possible. At this edge, the direct connection of nodes with each other is made possible by having access

to content locally whenever it is available [107]. Several different architectural approaches have been proposed to accommodate these network architecture changes. These approaches, which include TRIAD, CBCB, DONA, 4WARD, NetInf, PSIRP, MobilityFirst, SAIL, CONET, COMET, PURSUIT, NDN, and CONVERGENCE, each offer unique solutions and potential benefits for the future of the Internet [107, 50].

This literature review is a testament to the collaborative effort of researchers in shaping the future of the Internet, NDN, which is the Named Data Network. It presents a comprehensive overview of the related research areas implemented concerning the current subject matter. This chapter is structured as follows: In Section 2.1, the benefits and features of NDN are broadly classified into system architecture, system services, and NDN applications such as Video on Demand (VoD). NDN systems are serviced by forwarding, routing, security, mobility, and caching. Section 2.3 highlights that in-network caching is characterised by on-path caching and off-path caching, whereas on-path caching contains multiple types, as explained in this chapter. In addition, Section 2.4 presents the famous NDN cache placement mechanisms with VoD services. Section 2.5 compares caching placement mechanisms. Finally, the chapter ends with a summary presented in Section 2.6.

## **2.1 NDN Classification**

The NDN classification, as shown in Figure 2.1, is proposed based on an analysis of its architectural design, important auxiliary support, and key functional characteristics. The NDN features are broadly classified into the system's architecture, services, and applications of NDN [103]. The research briefly reviews various real-life applications previously developed using NDN and potential applications that can be developed in the future, with a particular focus on VoD due to its challenges with delay and Quality

of Service (QoS). These challenges underscore the efficiency of NDN in managing Internet responsibilities for various applications in the coming years.

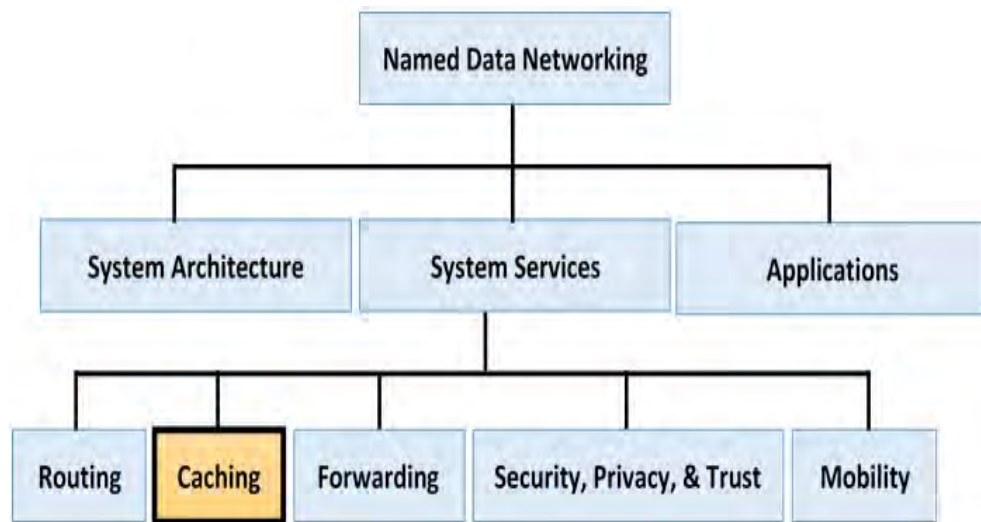


Figure 2.1. Classification of NDN

### 2.1.1 NDN System Architecture

This section discusses the organisation of various components, their working principles, and their interaction within the system. It's important to note that every NDN router plays a crucial role in maintaining three data structures: Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB). Each data packet in the NDN system can independently confirm its identity, ownership, and authenticity. As part of its responsibilities, the NDN router places a copy of the data packet in the CS, contributing to the optimisation of bandwidth, the enhancement of sharing probability, and the minimisation of content retrieval time. Searching CS entries follows the exact matching of names, character by character [50].

Moreover, PIT maintains an entry for every incoming interest packet until its corresponding data packet arrives. FIB has the function of significantly reducing network load and data latency by maintaining next-hop(s) and other relevant information for every destination name prefix that is reachable, such as the quality

of the link, the cost of the path, or the available bandwidth [108].

## **2.1.2 NDN System Services**

This section concerns the key efficient features of NDN. These features and characteristics include routing, caching, forwarding, security, and mobility. NDN supports in-network caching of the user's requested content. The following subsections discuss these major features of NDN in recent years.

### **2.1.2.1 Forwarding**

The NDN forwarding plane can find and fix failures on links, nodes, or packets, so routing does not need continuous FIB updates. This makes the NDN routing plane more stable and scalable. NDN forwarding is divided into scalable forwarding and forwarding mechanisms. Intelligent and stately forwarding is supported through scalable forwarding. This, in turn, allows NDN routers to measure the packet's Round-Trip-Time (RTT), packet losses, throughput, and alternative path during congestion and overhead. In addition, the forwarding plane in NDN simultaneously serves as a control plane because the forwarding mechanisms to accomplish all the decisions are required to forward the Interest and Data Packets [109].

### **2.1.2.2 Routing**

Routing is responsible for establishing topology and policies and handling long-term changes. It also updates the corresponding forwarding table. At NDN, its routing protocol synchronises with the NDN forwarding plane for probing and ranking the interface [110].

### **2.1.2.3 Security, Privacy & Trust**

Content security is a major and crucial requirement for any data-centric network [111]. One of the key problems facing NDN security is the cost-effectiveness of its security operations, management of trust, and protection of privacy. In NDN, a standard public key for security is used to sign every Data Packet, which is necessary for authenticity verification by anyone [50]. The important properties of trusted content are authentication of origin, data integrity, and data relevance against the requested data. An appropriate signature mechanism is selected, such as Rivest Shamir Adleman (RSA) [111, 112].

Trust management is used to authenticate a key assigned to a particular pocket in an application [50]. Trust performance can be average, good, or excellent [113]. To achieve privacy in NDN, the routers store the user's request. In NDN, encryption of both the content and its name is necessary for the provision of privacy, and this is similar to the Anonymous Named Data Networking Application (ANDaNA) [114].

### **2.1.2.4 Mobility**

The NDN's mobility help is often broken down into three areas. These three concepts are network mobility, customer mobility, and producer mobility. NDN enables network efficiency and prompt information transmission to these mobile consumers by organically supporting the mobile consumer's needs [115]. Hence, the consideration of NDN mobility raises two significant questions. The first question concerns consumer mobility in the sense of how the requested data can be returned to a moving consumer, and the second question is in regard to producer mobility in terms of how a consumer's interests can get to the generated data through moving producers, such as mobile [19]. In C. Fang et al. [116], mobility was classified as a subscriber, a publisher, and network mobility.

### 2.1.2.5 Caching

The practice of in-network caching, which involves storing content at intermediate nodes, is a fundamental aspect of the NDN data delivery model. This approach offers numerous advantages, such as reducing overhead on the producer side by separating content from its creators. It also eliminates single points of failure. In-network caching is particularly beneficial for dynamic content in multicast or retransmission scenarios due to packet loss. Most importantly, it relieves the network of unnecessary burdens, thereby enhancing performance, reducing network load, and decreasing data dissemination latency [117, 118].

The importance of measuring cache performance can give the listed metrics context:

1. The content retrieval delay is the time between generating and delivering the content request to the consumer. It is crucial for performance in time-sensitive applications. Faster data retrieval leads to lower cache latency.
2. The cache footprint, the amount of cache space a workload requires, is a powerful metric. Understanding and utilising this metric empowers optimising cache placement, ensuring that frequently accessed data fits within the available space. This control over cache space can significantly enhance the efficiency of the caching system.
3. The cache hit rate, a pivotal metric, directly influences data retrieval speed. A higher cache hit rate indicates that the cache effectively stores frequently accessed data, leading to faster data retrieval and reduced latency. This metric is often used as a primary indicator of cache effectiveness, and its practical implications on data retrieval speed are significant. Understanding the role of cache hit rate in data retrieval speed is crucial for network engineers, system administrators, and researchers in the field of computer networking, as it directly

impacts network performance.

4. Cache latency is the time it takes to access data from the cache once a request is made. Lower cache latency results in faster data retrieval, which is crucial for performance in time-sensitive applications.
5. Cache utilisation, a crucial metric, is the percentage of the cache actively used to store useful data. High cache utilisation indicates efficient use, reducing the likelihood of unnecessary evictions and cache misses. This metric plays a significant role in maintaining an optimally performing cache system, making it a key aspect to consider in cache management. Understanding and optimising this metric can greatly enhance the efficiency of the caching system, a fact that network engineers, system administrators, and researchers in the field of computer networking should be aware of.
6. The eviction rate is the frequency with which data is removed from the cache to make room for new data. A high eviction rate may indicate that the cache placement strategy could be more optimal, as it leads to more frequent cache misses.
7. The server load and store latency is the time it takes to load data from or store data in the cache. Lower latency indicates that the data placement strategy allows quick access and updates, reducing overall access time.

This study focuses on caching and will be briefly described in the following sections.

### **2.1.3 Applications**

NDN's architectural characteristics make it a versatile tool for various networking applications. Its functional features, including security, multi-path forwarding, name data routing, in-network caching, scalable forwarding, and infrastructure-less mobility, make it a promising alternative to the current Internet. This breadth of applications

underscores the potential of NDN in various fields [119, 120, 121]. The practical applications of NDN are vast and diverse, spanning multiple fields such as:

- NDN shows promise in Smart System Development (SSD), such as Smart Cities, Smart Grids, and the Internet of Things (IoT), where features like name data routing and in-network caching can improve the efficiency and security of data transfer between smart devices. For example, in a smart home system, NDN could facilitate secure and efficient communication between various IoT devices, allowing for seamless automation and control of household appliances. This smart home could include automatically adjusting thermostat settings, controlling lighting, or monitoring security cameras [122, 123].
- Building Automation Systems (BAS) can benefit from NDN's scalable forwarding capabilities, leading to more reliable and adaptable systems for managing buildings and infrastructure. For instance, NDN can enable BAS to efficiently control lighting, HVAC systems, and security cameras in large commercial buildings, improving energy efficiency and security measures. HVAC is a shortened form of the term "Heating, Ventilation, and Air Conditioning". A domestic HVAC system is a comprehensive home comfort system that can heat and cool your house while enhancing indoor air quality and regulating humidity levels. Additionally, NDN's ability to handle large amounts of data traffic improves the performance of intelligent grid systems, enabling better management of electricity distribution and consumption [124, 125].
- Using NDN in vehicular communication could revolutionise transportation systems by enabling secure, efficient data exchange between vehicles, infrastructure, and other devices on the road. For example, in a smart city setting, NDN could allow for seamless communication between autonomous vehicles and traffic lights, improving traffic flow and reducing accidents. Furthermore, NDN technology could also be utilised in emergency response systems to

quickly and efficiently relay critical information between first responders and command centres during natural disasters or other emergencies [126].

- Conferencing and educational systems stand to gain from NDN's multi-path forwarding feature, which can enhance the quality of real-time communication and collaboration among users. For example, in a remote learning environment, NDN could enable students and teachers to engage in interactive virtual classrooms with minimal latency and high-quality video streaming. Additionally, in business conferencing settings, NDN's ability to efficiently distribute data across multiple paths could lead to more reliable and secure communication between participants in different regions [127].
- NDN's potential applications in multimedia systems, gaming, entertainment, and VoD services highlight its versatility as a technology that could shape the future of content delivery and user experience across various platforms. For example, NDN could enhance multiplayer experiences in online gaming by reducing lag and improving overall performance, instilling optimism about the future of content delivery and user experience. In VoD services, NDN's ability to efficiently deliver high-quality video content leads to faster streaming speeds, better user satisfaction, and revolutionising entertainment [128].

This research proposal is particularly urgent as it focuses on VoD as one of the NDN applications. The unique need for big-size caching in CS in VoD is a pressing issue. VoD's direct viewing makes delivery time a crucial factor. Also, Section 2.4 talks about how various cache placement mechanisms might affect the performance of VoD in NDN and how important this research is for boosting QoS through better VoD services.

## 2.2 Classification of Caching Mechanisms

Caching mechanism types are classified based on the cache decision mechanism, a critical component in the caching process that decides whether to cache the data packet in the intermediate routers. These decisions significantly impact the caching process, as they are the key determinants of the efficiency and effectiveness of the cache system. As a result, caching mechanisms are broadly classified into two groups: cache placement, which determines whether to place the content on the network, and cache replacement, which decides if the content should be cached at the router. Understanding caching replacement mechanisms can shed light on the caching system's overall behaviour and performance, affecting the strategy of placing video chunks. Both cache replacement and cache placement mechanisms are crucial in determining a caching system's overall effectiveness and efficiency. They work together to manage how data is stored and maintained in the cache, influencing the performance of the cache in terms of speed, hit rate, and resource utilisation, as shown in Figure 2.2.

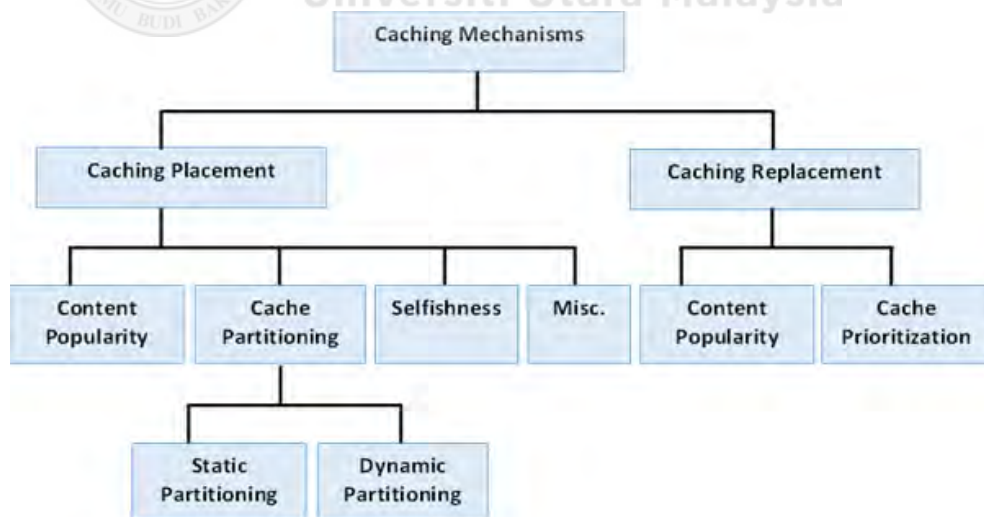


Figure 2.2. Classification of Different Caching Mechanisms

### 2.2.1 Cache Replacement

A very popular cache replacement mechanism is the Least Recently Used (LRU) mechanism, which discards contents that were least recently accessed. This mechanism possesses high performance and has the potential to increase cache hit probability through the storage of the most recent data for an additional time. Another important and well-known cache replacement mechanism is Least Frequently Used (LFU), which discards the contents that are the less frequently used first. The timing of the cache's decision can be based on the arrival of content at the router and the replacement of content. There is a further classification of cache replacement, which is based on the popularity of the content or prioritisation of content, such as the replacement of less popular and low-priority content being done first [129, 130, 131]. An understanding of caching replacement mechanisms can shed light on the overall behaviour and performance of caching systems, which in turn affects the mechanism of placing video chunks.

However, the cache's effectiveness is influenced by how well the placement and replacement mechanisms complement each other. However, poor placement mechanisms can lead to frequent conflicts, which can be mitigated by intelligent replacement mechanisms that minimise the impact of these conflicts. For example, if multiple pieces of data are mapped to the same cache location, they can displace each other, leading to conflict misses. Then, the replacement mechanism decides which of these conflicting entries to evict [132, 133, 134]. While understanding cache replacement is foundational, the primary challenge this study addresses is the coordinated caching placement for video chunks across nodes, as will be discussed in the following sections.

### 2.2.2 Cache Placement

The default inbuilt cache placement mechanism in the NDN architecture is Leave Copy Everywhere (LCE) [135]. In LCE, caching of the Data Packet is performed at all existing routers between the consumer and the producer, with the induction of significant cache redundancy. In other words, similar contents are cached at multiple nodes, reducing the diversity of the content in the entire system. Examples of already existent nine cache placement mechanisms for the reduction of cache redundancy are Leave Copy Down (LCD), Random Choice Caching (Rand), Probabilistic (Prob), Path Probabilistic Cache (PProb), Hybrid Caching (Cross), Centrality-based mechanism (Centrality) [72, 136, 137], Cache Placement Based on Compound Popularity Mechanism (CBCP) [67], and Cache Placement Mechanism Based on Entropy Weighting Method and TOPSIS (CBEWT) [138], as will be elaborated in detail later in this chapter.

While the LCE mechanism enables caching at all routers, more strategic cache placement mechanisms consider specific content characteristics. Storage of popular content is needed at the network edge to achieve a utility increase in cached content in the entire system. This increase comes with the advantage of minimising the latency of content downloading and improving the entire network's cache diversity. An efficient coordination scheme is also important among the routers to reduce redundancy and improve cache diversity [139]. Cache coordination schemes are of two types, which are explicit and implicit schemes, and that will be explained in the next Section 2.3 for a clear understanding of how the caching placement works to come up with cache placement mechanisms for VoD in Section 2.4.

The cache placement categories are divided into four major categories: content popularity, cache partitioning, selfishness, and miscellaneous, as shown in Figure 2.2.

- **Content popularity:** This concept involves caching only the most popular content on the network. The goal is to reduce traffic on the inter-domain, which refers to communication between different network domains or administrative entities. The total number of requests for the content, which remains stable over a short period, determines the popularity of the content. However, if large amounts of data, like VoD, are in the cache as a result of caching placement based on popularity, it may cause problems such as an occupied cache. In the NDN network, various types of traffic storage space exist in the content store of an NDN node. To efficiently share this limited storage space, cache partitioning is necessary.
- **Cache partitioning:** Cache partitioning refers to dividing the cache space into sections or segments, each dedicated to a specific type or class of data. The concept of cache partitioning has been researched in studies with the application of network traffic analysis, which increases the cache performance, like the cache hit ratio. To experience impressive cache performance, there should not be a replacement of application contents that can be reused, such as web and videos, with non-cache-able items like telephony apps, email, and others. Rather, the CS could be partitioned at the NDN router for the various types of traffic classes, such as Constant Bit Rate (CBR), which refers to a network that provides a guaranteed data rate, and non-Constant Bit Rate (non-CBR), which refers to a network that does not provide a guaranteed data rate, where the former is used for multimedia applications. Cache partitioning can be of two types: static or dynamic. One class cannot use another class's cache space in static partitioning. Meanwhile, in dynamic partitioning, the space of a class cache can be used by another traffic class when the former does not use the cache space at that specific time. Rezazad et al.'s [44] introduction of dynamic partitioning of the content store for NDN traffic uses the cache miss equation for dynamic partitioning of database buffers. This scheme minimises cache miss probability

and ensures fairness across various NDN traffic. Placement decisions can be optimised for specific content types, such as videos, by strategically segmenting the cache space. Furthermore, there is partitioning for CS [44, 140, 86] and there is partitioning for big data like video in multi-Cs that cache video types [141, 82, 42, 41, 142]. Partitioning video to multiple routers is called cooperative partitioning because the routers cooperate in migrating between themselves to enhance memory utilisation and increase hit rate and network footprint.

- **Selfishness:** considers every node to act selfishly. In the context of selfishness, caching refers to nodes making caching decisions based solely on their benefit rather than a collective network gain. This decision is because nodes will only coordinate the cache when there is a reduction in their accessible content cost. Such as a Not So Cooperative Cache (NSCC) scheme in which a self-centered node performs content caching if the coordination of the cache significantly reduces its own access cost through data access from either its local cache or its neighbour's cache. The potential for substantial cost reduction in the NSCC scheme offers a hopeful outlook for future content caching in computer networks. The main aim of this scheme is to discover which contents need to be cached at every node to minimise the individual node access cost [125, 143].
- **The miscellaneous:** It refers to a versatile category of caching mechanisms and offers a range of benefits in various contexts. For instance, it can cache game assets or levels for faster load times, improve streaming quality by caching video segments at edge locations, and facilitate ad hoc data distribution on mobile devices using the NDN network. The 'data interval' parameter, which determines the validity of cached data, is a key feature of miscellaneous caching and is typically an integer value in the data packet, routing, and PIT. However, Zeng et al. [144] have introduced the Ad-hoc On-Demand Distance Vector (AODV) caching scheme, a promising development in caching mechanisms. This scheme sets an integer data interval parameter in the data packet to

distribute contents along a returning path. The parameter plays a crucial role in determining the cached data interval along the interest-forwarded path, which is the data provider of the content sets. Each time content is forwarded downstream, the data interval value is decremented by 1. When the interval value becomes 0 at any intermediate node, CS caches the content and resets the data interval value. This process is repeated until the consumer receives the data packet. It's important to note that using the AODV caching scheme's data interval parameter significantly improves the effect of cached data distribution on mobile ad hoc NDN network performance [144, 145]. In addition, some authors have introduced caching to the routing and PIT to enhance the performance of the NDN forwarding daemon. Choi et al. proposed a Coordinated Routing and Caching (CoRC) scheme. This scheme is designed to mitigate the impact of routing scalability and boost in-network caching efficiency, offering the promising potential for improved network performance [125, 146].

### 2.3 In-network Caching

Traditional caching schemes used in network architectures include in-network, hierarchical, and proxy caching, each with a unique design. However, the practical benefits of in-network caching have captured the attention of researchers, making it a more promising area of study than traditional caching methods. Internet architectures, specifically content distribution and data delivery systems, employ hierarchical caching to arrange caches in a multi-level hierarchy, positioning different cache levels at various network points. These points range from the core network to the edge, and each serves specific roles in data storage and retrieval. Furthermore, proxy caching is a traditional Internet caching technique in which a proxy server stores copies of frequently requested data or web content to serve future requests more efficiently. In addition, in-network caching refers to the ability of network nodes to store copies

of data that pass through them temporarily. Examples of these nodes are routers, switches, or edge devices. Subsequent requests can then directly receive this cached data, eliminating the need to fetch the same data from the source repeatedly. However, in-network caching is a key feature of NDN that is crucial in enhancing the efficiency and performance of data dissemination in Smart Systems like Smart Cities, IoT, and Smart Grids. Therefore, this study proposes a caching mechanism that relies on on-path caching, which then enhances collaboration and content sharing among routers. This mechanism is suggested for storing video in the routers and reducing End-to-End Delay (E2EDelay) effectively in NDN for VoD services.

In-network caching focuses more on critical network operations. NDN caching at network routers presents an in-network caching challenge. Because in-network caching works differently from Web and hierarchical caching, many techniques and approaches have been studied and suggested as possible solutions. Previous studies have demonstrated that in-network caching has a significant impact on network performance enhancement, throughput growth, traffic, and retrieval time reduction [147, 139]. This collective acceptance of in-network caching as an NDN enhancement underscores its vital role in network design. Note that in-network caching is a vital design component, and there are two types: off-path caching and on-path caching. They refer to where the cache is located relative to the data flow between a client and a server. Off-path caching is an in-network concept that stores cached data in network nodes that do not directly connect to the data source or the requester. It stores data at alternative or secondary nodes that are not immediately involved in handling the original request. On the other hand, on-path caching caches data at nodes along the route of the data request. However, Figure 2.3 explains and shows various caching schemes.

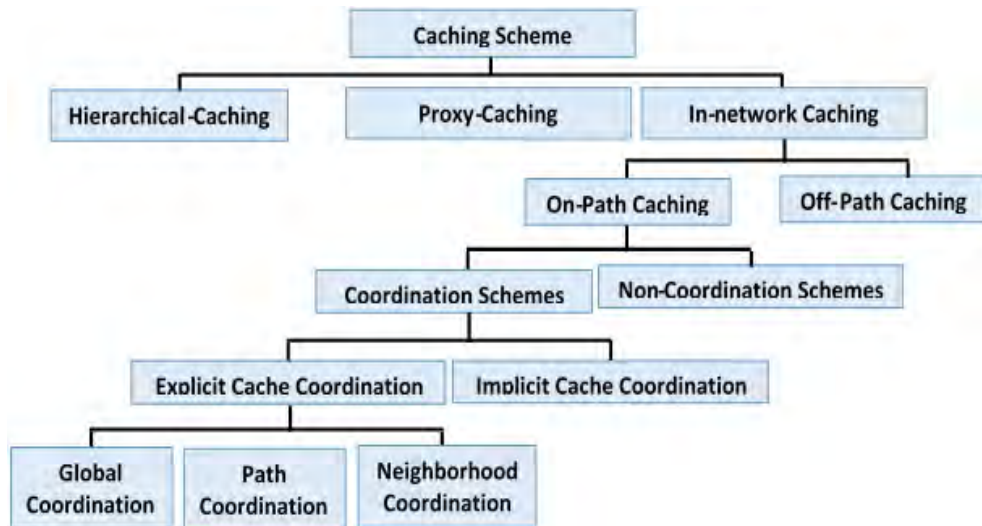


Figure 2.3. Caching Schemes

### 2.3.1 Off-path Caching

Off-path caching is a mechanism that does not directly connect to the communication path between the user and the server. It is located separately, requiring extra steps for the client to retrieve data. Off-path is useful for scenarios where not all data needs to be cached or different data caching strategies are required for different content types. Managing off-path caches can be complex, requiring additional mechanisms to ensure proper access. In scenarios where different data caching strategies are required for different content types, it proves useful. Therefore, implementing off-path caching can become more complex, leading to increased hop counts. Accessing data from an off-path cache can introduce additional latency compared to on-path caches, especially if the data is VoD [148, 149, 150, 151]. Therefore, this study proposes a caching mechanism that relies on on-path caching, which then enhances collaboration and content sharing among routers. This mechanism is suggested for storing video in the routers and reducing E2EDelay effectively in NDN for VoD services.

### 2.3.2 On-path caching

It is a caching technique that is commonly utilised in NDN. It caches the data directly on the path between the client and server, which leads to reduced latency and retrieval time. On-path caching behaviour is displayed when interested items are being served, as the data chunk will be cached at each router along the path from the content producer to the consumer. At the same time, the interested items are cached at PIT, while the data chunk is cached in different CS(s). In addition, the usage of network bandwidth is reduced, especially for frequently accessed content.

Consequently, the capability of the caches is investigated as a specified requirement that permits sufficient content to be available when requests are submitted. For example, the LFU caching replacement mechanism caches the content depending on the frequency of requests. In contrast, the LRU caching replacement mechanism caches content based on recent use [131, 152]. As the provision and intelligence of flooding content become less complex, more features are necessary for on-path caching [153].

For NDN, maintaining caching ability is required from every router, referred to as in-network caching. This behaviour of on-path caching is advantageous in several ways [9], such as:

- **Multi-cast support:** NDN gives the routers the ability to cache. Edge routers can now handle content distribution rather than the content's original creators. This support goes a long way toward saving massive resources in the network.
- **Chunk re-transmission:** The request must be re-transmitted to achieve reliable transport protocols. Better efficiency will be recorded if the contents are re-transmitted from closer routers to the users than if re-transmitted from the

originating servers in NDN.

Although on-path caching is beneficial for distributing content, it still possesses certain shortcomings. The major disadvantage attributed to on-path caching is space wastage because it keeps similar contents at multiple routers. This space problem occurs because data caching at each router on the delivery path can store similar replicas all over the network, specifically for the contents that have higher popularity. Due to this, it is often believed that on-path caching wastes space and simultaneously lowers network speed by reducing the global hit ratio due to unpopular content. In addition, scaling on-path caches can be challenging due to handling all the traffic passing through the path, requiring significant resources. In research to improve the functionality of in-network caching in NDN, several caching techniques have been created in response to these issues [139].

In NDN, two kinds of on-path caching schemes are coordinated and non-coordinated. The next subsections will introduce these various caching schemes and pertinent research using them [139].

### **2.3.2.1 Non-coordinated Caching Policy**

Non-coordinated caching refers to routers operating a cache independently while upholding their regulations. Each router independently operates the canonical caching strategy, and the typical methodology is based on frequency, request, or past usage [154]. Non-coordinated caching needs help to make caching decisions based on global information, making it nearly impossible to maximise the caching space properly. NDN adopts on-path caching, such as LCE, and Rand [155, 156, 157, 158]. More details of these mechanisms are explained in Section 2.4. Researchers Chai et al. [159] emphasised that distributed caching decisions in non-cooperative caching will only

partially exploit the potential advantages that in-network caching can bring. Therefore, this non-coordinated caching policy leads to waste or under-utilisation of the cache space.

On the other hand, this study focuses on VoD services, which require maximum cache space utilisation. Thus, this study proposes a coordinated cache placement mechanism for VoD services. The next section will explain the additional advantages of coordinated caching.

### **2.3.2.2 Coordinated Caching Policy**

Coordinated caching refers to the routers not working independently but putting out a joint effort to store different content. From access to the shared information between routers, the coordinated scheme can make caching decisions concerning precise performance objectives. Coordinated caching achieves relatively better utilisation of caching space than non-coordinated caching. As a result of the limited caching capacity, there is no maintenance of unpopular content in such conditions, and there is a probability of reducing the global hit ratio being experienced. Another concern that is raised deals with the fairness issue, which seeks an answer to whether every content possesses an equal right to benefit from the capacity of the cache. Likewise, more traffic overhead will be generated within the network due to the high volume of information exchanged between the routers. Choosing a criterion for making a caching decision may be costly sometimes, bringing about impracticality in deploying policies. Therefore, coordination schemes are of two types: explicit and implicit cache coordination schemes.

- **Explicit cache coordination schemes:** In this scheme, it is necessary to have prior knowledge of the content access frequency, the topology of the cache network,

and the state information of the network cache. This scheme is required to be able to make placement decisions about content. Explicit coordination allows for more strategic placement and retrieval of content across caches. By coordinating what is stored where, the system can reduce redundancy and better use available storage and bandwidth. In addition, the system administrators can design caching strategies that are predictable and optimised for specific workloads or traffic patterns, leading to more consistent and reliable performance. It can distribute the load more evenly across the network, ensuring that no single cache is overwhelmed while others are underutilised. A centralised or coordinated control mechanism allows for easier enforcement of policies, such as content prioritisation or security protocols. By explicitly coordinating which content is cached and where, the system can ensure that high-priority or frequently accessed content is readily available; the overall QoS for end users can be improved. Thus, explicit coordination can optimise resource utilisation, predictable performance, load balancing, centralised control, and improved QoS. This scheme is essential in environments with strict performance or compliance requirements. This scheme helps maintain a balanced and efficient system, especially under high-traffic conditions [160, 161]. Concerning making these decisions, this explicit coordination scheme can be further classified to work in three forms:

1. **Global coordination:** refers to all the cache nodes involved in the decision. It is ideal for networks requiring comprehensive control and optimisation across a wide area, focusing on maximising resource efficiency and consistent policy enforcement.
2. **Path coordination:** means that the cache nodes along the path between the producer and consumer are involved in the decision. It reduces latency and optimises bandwidth usage, making it well-suited for scenarios where content needs to be delivered quickly and efficiently along specific routes.

3. **Neighborhood coordination:** involves all the cache node neighbours in the decision. It provides localised optimisation and fault tolerance, making it a good choice for networks where regional load balancing and adaptability to local demand are critical.

This explicit approach is typically accomplished by inserting the required information for coordination into the requested content. For example, the needed information may either be each cache node's status or the frequency of the requested object at every cache node. When a request is received, the hit node will use the obtained information to compute the optimal placement mechanism for content. For example, a centrality-based caching mechanism exploits the concept of (ego network) betweenness centrality to improve the caching gain. On the other hand, betweenness centrality means that the content is only stored at the nodes that have the best chance of getting a cache hit along the path of delivery [159, 161], as explained in Subsection 2.4.7.

- **Implicit cache coordination schemes:** In this scheme, every cache node does not need to have prior knowledge of the status of the other cache nodes. This scheme suggests that the nodes only exchange information before deciding where to store the material. It is flexible because decisions are made based on local conditions such as user demand, hit rates, or network traffic. This flexibility allows the system to adapt to varying conditions without central oversight. It has scalability since there is no need for centralised control or frequent communication between caches, and implicit coordination scales well with large, distributed networks. Each cache can manage its operations without overburdening a central controller. It reduces overhead because implicit coordination minimises communication overhead, and caches do not need to constantly exchange information or synchronise their actions. This scheme can lead to faster decision-making and reduced latency. With implicit coordination, there is less need for complex management protocols or coordination algorithms,

which can simplify the overall system design and implementation [161].

There are many implicit placement mechanisms, such as LCD, Prob, Content-Aware Caching (ProbCache) [162], and PProb. Furthermore, there is Cross, CBCP, and CBEWT [163, 164, 16, 100, 67, 138]. Due to its flexibility in making decisions based on local conditions, each implicit coordinated cache placement mechanism above is decided based on different conditions, as explained in Section 2.4.

Based on the previous considerations, cache coordination is divided into explicit and implicit categories according to the level of autonomy in cache decision-making. The study's specific goals determine whether implicit or explicit cache coordination is used. Thus, this study's goals are to place VoD services in the cache to reduce E2EDelay and server load, raise the average hit ratio and footprint, and improve memory utilisation. However, explicit coordination is ideal for environments where performance predictability, resource optimisation, and strict QoS requirements are critical, such as VoD systems or CDNs. Conversely, implicit coordination is generally more suited for large and dynamic networks where scalability, flexibility, and fault tolerance are prioritised. Therefore, the explicit cache coordination scheme using path coordination is more appropriate for VoD because it provides better control over the caching process, improves performance, reduces latency, better utilises caching space, and provides a more reliable user experience.

## **2.4 Cache Placement Mechanisms and VoD Services**

A cache, which instantly saves the accessible data, is quick-access memory. Caching has been studied and employed in several computer-related fields, most notably in NDN. The caching placement mechanisms determine the sequence of routers from the actual content source to the user. Operational cache placement mechanisms are

significant for reducing Internet Service Providers (ISPs) traffic and getting good performance in NDN. Recent cache placement mechanisms are explained below to determine which achieved the best delivery time during VoD services [165, 166, 167, 168, 106, 169] to design a new cache placement mechanism in NDN during VoD.

#### **2.4.1 Leave Copy Everywhere (LCE)**

LCE is an extensively utilised caching mechanism in NDN. It's non-coordinated cache placement. Thus, it does not require a control parameter to decide whether to cache or not. It is stable due to its simplicity, inherent redundancy, and lack of coordination overhead. By caching content at every node along the path independently, LCE avoids complex synchronisation issues, provides fault tolerance, and maintains content availability even in the face of network failures or node issues. This straightforward approach contributes to the overall stability and robustness of the caching system. Additionally, Li et al. [170] highlighted that LCE can lead to excessive duplication of content, especially across each router on the content's delivery path. However, this means that when the content is requested by different users, the number of nodes holding a copy of the content increases, leading to a larger content population within the network. Therefore, the content population is increased, and the utilisation of caching is expanded. In the case of LCE, the output stays stable. If the system anticipates increased demand due to emergencies like flash floods, LCE can be advantageous because it ensures that content is cached at multiple points without needing advanced coordination [168, 169].

This mechanism was designed to cache within multi-level routers in networks. As shown in Figure 2.4, the process is initiated by the user requesting the available content on the server. Next, the request gets forwarded through routers till it arrives at the publisher. For LCE, the occurrence of a hit either at a Level  $L$  cache or the main server brings about the storage of a copy of the content that was requested in all in-

between nodes (levels  $L - 1, \dots, 1$ ) on the route, that is, beginning from the hit node through the user that made the request. As a result of the contents being cached at all nodes in between, when there is an arrival of new content, this content is forwarded to the intermediate node rather than the server. With a copy of the new content, this node sends the data reply. The concept of content caching at every router can improve bandwidth utilisation because the requests are serviceable locally rather than being sent to the server. Although the content replicas are not useful, the intermediate node's cache will be filled up, and the contents just arriving will not be able to contribute to the cache hit.

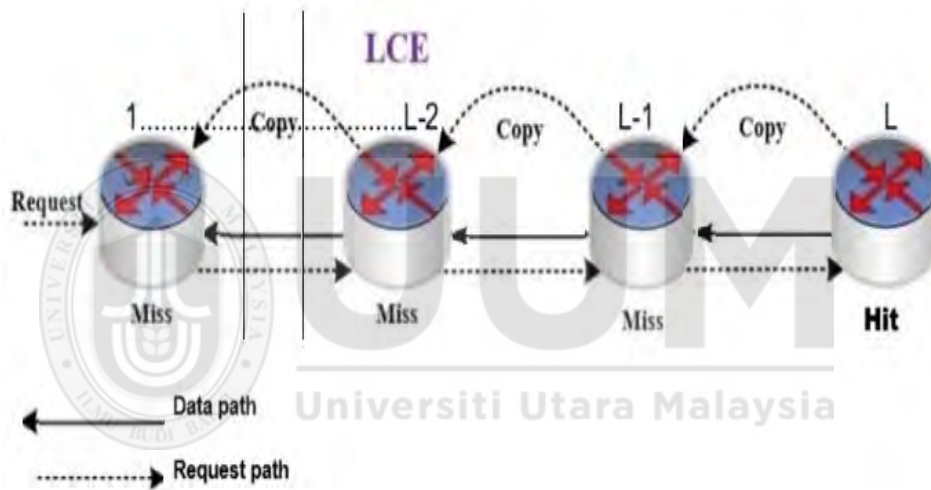


Figure 2.4. The Operation of LCE

There is this impractical notion that content caching at all nodes will bring about an increased cache hit ratio. This idea is said to be unrealistic because, at the point where a node's cache overwhelms, and there is an arrival of new content, it becomes expedient to make a replacement of one of the cached contents. Replacement of content mostly follows the replacement mechanism of LFU or LRU mechanism [171, 168].

On the other hand, the LCE mechanism shows a performance disadvantage when processing resources on multiple platforms. Putting a copy in each router on the path

leads to utilising all the capacity of the content store, applying plenty of the cache replacement mechanism, and spending more actions to serve the same request. These simultaneously cause spending more time than the other cache placement mechanisms, making their delivery time higher than theirs, especially when the content is VoD due to the large size of the video. For this reason, there are other better mechanisms than LCE for caching placement.

For video streaming or VoD services, the LCE approach results in multiple copies of the same video content being stored across the network. This redundancy can lead to inefficient use of cache storage, especially for less popular or niche content that may not justify widespread caching. Managing a large number of redundant copies can become cumbersome, potentially leading to issues with cache storage and maintenance. Since LCE is non-coordinated, there is no central mechanism to manage or update caches. Thus, it can lead to inconsistencies in the content available at different nodes, particularly if videos are updated or removed. In addition, cache eviction policies might be inconsistent across nodes. Some nodes may store outdated or irrelevant content, while newer, more relevant content needs to be cached effectively. For videos that are rarely requested, the LCE strategy may result in inefficient use of resources. Caching such content at every node can lead to wasted storage space and potentially unnecessary network traffic. There might be increased network traffic due to the distribution of content. Thus, it could lead to higher bandwidth usage and potential congestion, especially during peak times or for high-definition content [168, 106].

LCE's non-coordinated cache placement offers several advantages for VoD services. Caching can be effective in certain scenarios, particularly for ensuring high availability and reducing latency; its disadvantages include high storage requirements, increased cache evictions, poor scalability, and inefficient use of bandwidth and cache resources.

These drawbacks make LCE less suitable for environments where resources are constrained or where optimising storage and network efficiency is a priority. Therefore, LCE is often used selectively or in conjunction with other caching strategies that can mitigate these issues.

#### **2.4.2 Leave Copy Down (LCD)**

Cache placement implicitly coordinates with the LCD caching placement mechanism. The control parameter is the content location in the previous level cache [169]. Figure 2.5 illustrates how it allows only one router to save a new copy of the interested object. LCD selectively caches content based on its previous cache location, unlike LCE, which caches content at every node. This can result in a more effective utilisation of cache resources. This approach avoids unnecessary duplication of content across the network, optimising storage usage. Therefore, LCD reduces the number of redundant copies in the network compared to LCE. By freeing up cache space for other content, we can cache a wider range of videos closer to users [106].

To avoid numerous replacements and repetitions of caching similar objects at multiple levels, apply this LCD mechanism. The term multi-caching levels refers to a hierarchical or layered approach to caching within a network. The LCD structure enables the network to store content at various levels or locations, contingent on factors such as demand, user proximity, and available resources. For example, consider a VoD service with a three-level caching hierarchy. When a local area user requests a video available in the regional cache ( $L$ ), but not in the edge cache, the system delivers the video to the user and simultaneously caches it at the edge cache ( $L - 1$ ). Also, the LCD can work better under interconnection topologies and numerous workloads, particularly while studying the caching behaviour of multi-caching levels [172]. In the context of VoD services, LCD can offer several advantages and challenges [106, 120].

It was primarily created for hierarchical web caching approaches and included sophisticated caching references if a cache hit is only performed for a specific piece of content along the path. This mechanism primarily benefits from the ease of replicating content to fit a specific cache size in a specific path. As a result, data becomes more visible as it moves closer to the network's edge. This mechanism's drawback lies in the potential failure to transmit a cache reference to the source when writing data across multiple platforms [173, 174].

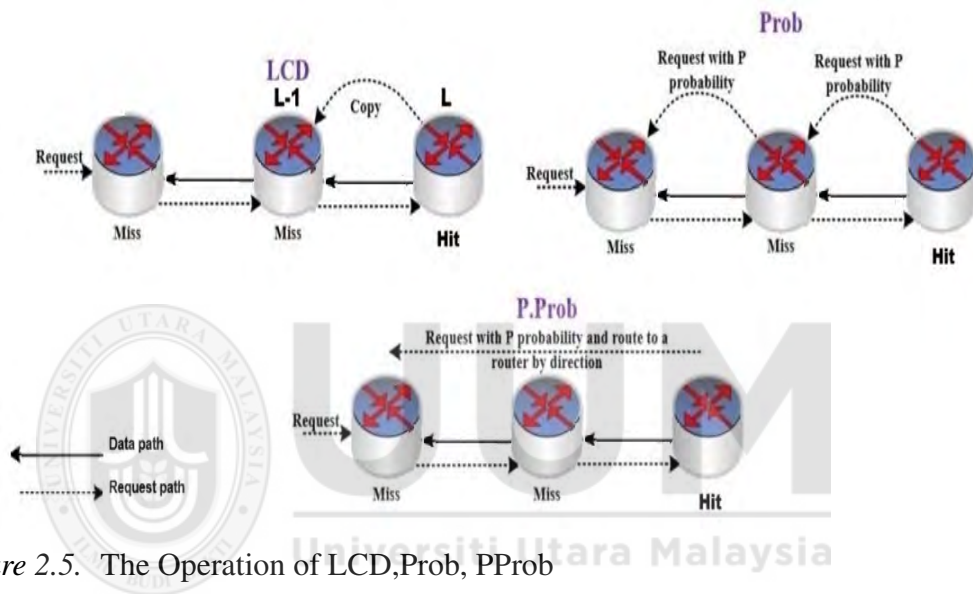


Figure 2.5. The Operation of LCD, Prob, P.Prob

Therefore, LCD has less redundancy along the path, fewer operations for each request, more simplicity, a suitable cache size, and more moderation. All these advantages of LCD result in a shorter delivery time compared to LCE, due to two key factors. First, when increasing the workload, LCD strategically places content closer to users, reduces unnecessary duplication, optimises network bandwidth, and adapts more effectively to user demand. The second reason is that when increasing the workload, LCE places the content in every router in one request, then the cache will fill in this case and need more replacements in each request. This situation intensifies the difficulty in locating the same request, as it undergoes frequent replacement with each new request. Furthermore, in this case, it results in multiple additional processes that increase the

delay.

Moreover, by focusing on selective caching and reducing the distance content needs to travel, LCD ensures quicker access and more efficient content delivery, especially in scenarios with high demand, varying access patterns, and large sizes of video, particularly VoD. Since caching of contents only occurs at a single node (that is,  $L - 1$  level node), LCD can increase the cache hit ratio and utilise the cache space effectively. However, in a case where there is a frequent request for some content through the use of popular content, the cache space will eventually become crowded. As a result, there will be no more available space for the new content. Crowding occurs when all nodes on the link begin to cache popular content. This is why the LCD also uses LRU to replace content in scenarios where the cache is full. This creates an excessive eviction operation in which newly arriving content replaces popular content. Accordingly, all the new requests replacing the evicted contents and then forwarding to the server maximise bandwidth utilisation. Despite the implicit LCD coordination, the strategy ensures that only the most relevant nodes cache the content. This leads to better cache management. This adaptability ensures that VoD services can efficiently respond to changing viewing trends. By caching content at the node closest to the user who requested it, LCD ensures that subsequent requests can be served with minimal latency, faster start times, and fewer buffering interruptions during video playback; the network can deliver these data-intensive streams effectively; LCD reduces the overall bandwidth consumption as fewer nodes are involved in caching the same content, and the probability of network congestion can decrease, particularly during peak viewing times.

Since LCD does not cache content at every node along the path, there's a higher probability that network congestion can decrease, particularly during peak viewing times of cache misses if a different user requests the same content from a different part

of the network. Such requests may experience a slight increase in latency due to the need to fetch content from a more distant node. LCD's effectiveness is dependent on content popularity. If users do not frequently request content, the cache may not hold it close enough to future users, potentially leading to inefficiencies for less popular or niche content. Although LCD reduces the need for broad cache coordination, it still requires intelligent eviction policies to ensure that the most relevant content remains cached. As the system requests and caches new content closer to the user, it may need to evict older content, a process that can be complex to manage effectively. The system must strike a balance between maintaining the cache of popular content and making room for new content. Improper management could result in the removal of content that is still necessary, leading to unnecessary cache misses. As the number of users and content requests increases, the LCD strategy must scale effectively to maintain its benefits. This may require careful tuning of caching parameters to avoid overloading downstream nodes with too much content. In scenarios where regional traffic patterns vary significantly, LCD might struggle to cache content efficiently across different parts of the network, potentially leading to uneven performance across regions. The network topology closely influences the efficiency of LCD. In networks with complicated or non-linear topologies, it can be hard to find the "downstream" node and cache content in the best place, which could make the strategy less effective. In VoD services, the path length from the content source to the user can vary. In cases where the path is long, caching only at the closest node may not be sufficient to reduce latency for users further away from the cached content. Therefore, when a video is not popular and the user's path is lengthy, an increase in the cache size can lead to a delay. This is because the cache size may not fill quickly, resulting in a lower replacement rate for the content. This lower replacement results in the video remaining on the same long path for an extended period. At the same time, not having a popular video does not help to make the content closer to the user with each request [174, 120, 6, 168, 175].

LCD's implicit coordinated cache placement offers targeted efficiency, reduced redundancy, and improved user experience for VoD services. By caching content closer to the end-users who request it, LCD reduces latency and optimises the use of cache resources, making it well-suited for delivering high-demand VoD content. However, the strategy also presents challenges, including potential cache misses, eviction complexity, and scalability concerns. In practice, LCD might be most effective in scenarios where content popularity is predictable and network topology supports efficient downstream caching. Properly managing cache eviction policies and adjusting to changing user demand are crucial for maximising the benefits of LCD in VoD services.

### **2.4.3 Random Choice Caching (Rand)**

It is a non-coordinated caching placement mechanism in NDN because it does not require a control parameter to decide whether to cache or not. A randomisation process independently determines each caching placement. Furthermore, cache sets, cores, or nodes do not directly communicate or coordinate to determine the placement of content. This content item is cached randomly in one router along the shortest delivery path. The Rand caching mechanism is highly viable across different hardware, software, and networking platforms. Its core strength lies in its flexibility and adaptability, allowing it to improve cache performance in a wide variety of environments. Whether implemented in hardware systems, managed by software, or deployed across network platforms, Rand provides consistent benefits by reducing conflict misses, balancing load, and optimising data access across diverse configurations. This broad applicability makes Rand a robust and effective caching solution for many different types of platforms [176, 161, 106].

Despite encountering some challenges when handling packets across various platforms, it is highly stable in peer-to-peer network activities because of its

decentralised nature, scalability, robust load balancing, fault tolerance, and design simplicity. By distributing cached content randomly across the network, it ensures even distribution, resilience to node dynamics, and reduced impact from node failures, all of which contribute to the overall stability of the system [177, 176, 156].

The advantage lies in its ability to support multiple platforms' node numbers [178]. Although this mechanism typically performs poorly in terms of cache hits, it serves as a baseline [179, 106]. Randomly, it identifies a content resource on the delivery path where it will cache a data packet. As a result, the multi-layer processing nodes delay the packet's transmission from one location to another.

This approach has important implications for VoD services due to the nature of video content, which requires large files and high bandwidth demand. By avoiding complex coordination overhead, it scales well in large networks, making it efficient. Additionally, it lessens the signalling load as each node stores information independently, eliminating the need for communication with others.

On the other hand, the Rand placement mechanism may lead to multiple nodes caching the same video chunks, which results in inefficient use of cache space. Retrieval latency increases due to the need to fetch content from distant nodes or the source. Therefore, this higher latency leads to potential packet loss, which may force the system to lower the video quality to maintain smooth playback, impacting the overall service quality. Since VoD services involve a large amount of data, the chance of hitting a cached copy under RAND can be lower compared to more optimised approaches. This lower hit ratio can affect user experience due to slower content retrieval times. Due to the randomness of the caching decisions, some nodes may end up serving a disproportionate number of video requests, leading to uneven load distribution and potential bottlenecks [176, 161].

RAND Non-coordinated cache placement in NDN for VoD services offers scalability, simplicity, and fault tolerance but faces significant challenges in terms of cache efficiency, load balancing, and content retrieval latency. While random caching works well in highly dynamic or large networks, its lack of coordination can lead to suboptimal caching for VoD services, where user experience and low-latency delivery are critical. To improve performance, a coordinated mechanism that caches content on the reverse path near the consumer may be necessary.

#### **2.4.4 Probabilistic Cache (Prob)**

It is generally considered an implicit coordinated caching mechanism in the context of Named Data Networking (NDN) because the nodes in the network do not communicate directly or explicitly with each other to make caching decisions. However, there is a shared underlying mechanism that guides how content is cached across the network, which creates a form of indirect or implicit coordination [173, 16, 162]. This caching mechanism copies the request's content at every network node along the delivery path from producer to consumer, with a specific probability  $P$ . However, when the probability  $P$  equals 1, this mechanism begins to resemble the LCE mechanism [35, 161, 16, 156]. To clarify, each network node will receive a copy of the content, as illustrated in Figure 2.4.

Prob [58, 180] is a randomised version of LCE because they have similar operations where every node can cache a copy of the requested content on the path from the hit server or node to the requesting host. Note that a node in the middle of the network does not cache a copy with probability  $1-P$ ; rather, it only keeps a local copy with probability  $P$ . Therefore, a node caches content with a specific probability. We can model this process of probability as a Bernoulli trial, which determines whether to cache content at a node. The Bernoulli trial, often known as a random experiment, has two outcomes: failure and success. A Bernoulli random variable accepts values of 0 or

1 depending on whether the content is accessed or not, with a hit being a success and a miss being a failure [181, 182, 106].

These Bernoulli trials affect key performance metrics such as caching efficiency, cost, latency, availability, scalability, and robustness. By adjusting the probability  $p$  of caching decisions, the network can find a balance between efficient cache utilisation, low latency, and reduced network traffic while keeping storage and network costs under control. Higher values enhance cache availability and reduce latency, albeit at the expense of increased storage and potential resource overuse. Lower values conserve cache space, but they can escalate cache misses and network traffic, resulting in increased latency. By fine-tuning the caching probability, Prob ensures that caching is both efficient and scalable, which is particularly important for handling the high-demand, dynamic nature of VoD services.

Prob attempts to minimise redundancy between network caches. Therefore, the probability of finding content along the path will be higher for subsequent requests. However, it is more stable in distributed networks. The hop between the user's router and the publisher will be lower in Prob [117, 72]. At the same time, hop is the distance to the closest node that contains a copy of the content object [35]. This leads to a short distance between the user and the server, low redundancy, and less delivery time. This hop decrement diminishes because it often replaces contents at nodes close to the user(s). This reduces network congestion and overall bandwidth consumption. Nearby caches can retrieve content instead of repeatedly fetching it from distant origins, particularly for popular videos in VoD services.

In the investigation of Psaras et al. [16], duplication in caching is a dangerous problem in localised and real-time data publishing in router caches. The researchers emphasised that the Prob mechanism can reduce caching redundancy by about 20% on server hits.

Approximately 8% of the total number of hops is required to access the cached data. It can handle high packet loads on multiple platforms, process a variety of nodes, and is stable in most network settings. Prob's drawback is the potential for delay when the packet elements do not align with the linked path [16, 162, 72, 156].

As a result, the Prob mechanism is efficient at decreasing network traffic redundancy by a significant amount, which can lead to a reduction in delay time compared to other placement mechanisms. While in the context of VoD services, Prob implicitly coordinated cache placement offers simplicity and low overhead. For better performance, more adaptive and explicitly coordinated caching mechanisms are required to be proposed with better cache utilisation and less bandwidth.

#### **2.4.5 Path Probabilistic Cache (PProb)**

Prob is an implicit coordinated cache with one factor, while PProb is a more complex implicit mechanism with two factors. It coordinates caching by changing caching decisions based on how far away the nodes are from the origin server and how much shared storage space they have along the path. This way, nodes behave in a coordinated way without having to communicate with each other directly. In other words, the probability of caching in a router is dependent on the hop distance between the router's shared storage capacity and the source server [16]. It is crucial for content caching to occur as close to their destination as possible, as this increases the probability of maintaining some caching space at the network core for minimal content flows. This mechanism efficiently utilises network resources, reducing redundancy in the cache and therefore reducing network traffic, as shown in Figure 2.4. The optimal mechanism initiates a cache miss when it receives a requested item. Upon requesting an item from a specific packet again, the PProb patiently awaits the validation of the reference before storing the requested item's data in the cache or ending any processes [156, 161]. Hybrid caching offers much more efficient resource utilisation and higher cache hit

ratios than non-coordinated caching mechanisms like Rand, where content is cached randomly without considering path length or popularity.

As a result, this mechanism has the advantage of fewer missed opportunities when requesting products. Additionally, it minimises the time required to request duplicate items later. It supports multiple platforms, and the reference sequence for each node is consistent with the packet. The downside of this mechanism is that it may require more space to store things in order at various locations across the cache [183].

However, by caching content based on the node's position along the path, PProb achieves better bandwidth efficiency, reduces latency, and optimises cache space usage, making it particularly well-suited for the large-scale, high-demand nature of VoD services. Implicit coordination guarantees a distributed yet effective content cache, obviating the need for direct communication between nodes and enhancing the scalability and efficiency of the mechanism. However, PProb may struggle to handle highly dynamic content popularity patterns, whereas more adaptive, popularity-aware mechanisms might outperform it. For highly dynamic and rapidly changing content demands, such as trending videos, PProb may not cache popular content fast enough at the optimal nodes.

With larger cache sizes, PProb effectively manages the packet queue based on the node's number and priority [184, 170]. The researcher concludes that PProb is still considered acceptable when the cache size is 10GB. However, processing multimedia sources with larger cache sizes results in an increased delay [174]. Therefore, there is still a need to investigate a new placement mechanism to deal with high cache sizes, such as VoD. In some cases, explicitly coordinated caching mechanisms may provide slightly better performance because of the cost of increased complexity and communication overhead.

#### **2.4.6 Hybrid Caching (Cross)**

Repeated content in the cache is one of the most serious caching mechanisms today. It usually exists in cached data on many routers. Cross addresses the limitations of earlier mechanisms like PProb and Prob by integrating multiple strategies, such as path- and popularity-based caching, to create a more adaptive and efficient system. While PProb and Prob rely heavily on path distance or probabilistic decisions, this process depends on the content's popularity and network architecture. The researchers emphasised that a cross-layer scheme better minimises network traffic and increases the total cache hit rate [100]. Thus, it is a powerful implicitly coordinated caching mechanism in NDN that efficiently balances multiple factors, a function of the router's betweenness and the content popularity, to determine where the router caches an object [72, 106, 185].

The key benefit of this mechanism is its ability to configure the content store of the router to have the same size for various routing technologies. This capability is particularly advantageous in heterogeneous networks, where varying routing technologies could otherwise lead to inefficiencies in caching and data delivery, especially in VoD services. When receiving content from other nodes [185], Hybrid Cache selects small data. As a result, it has more delay, especially when the items have a lot of memory [35, 186]. The Cross mechanism has high delay performance in cache sizes of about *1GB*, *10GB*, *100GB*, and *1TB* [174].

#### **2.4.7 Centrality-based Mechanism**

While Pprob and Cross lack explicit cache placement mechanisms, the centrality-based caching mechanism is a highly efficient, explicitly coordinated caching strategy in NDN that optimises video delivery for VoD services by placing content at strategically important nodes based on their centrality in the network. A node with high degree centrality in an NDN network has many direct links to other routers or nodes, making

it more likely to handle multiple data requests.

In NDN, interdependence centrality can be understood in the context of an ego network—the network made up of a node (ego) and its direct neighbours (alters). If the node (ego) can only interact with the broader network through a small subset of these neighbours (alters), its interdependence centrality is high. In other words, if the node has many independent paths or connections that don't rely on a few specific neighbours, its interdependence centrality is low. In addition, the egocentric network is the structure consisting of the ego node, its immediate neighbours (alters), and the connections between these neighbours.

However, to increase cache efficiency and eliminate uncertainty about the effectiveness of the oversimplified random caching mechanism, this node relies on centrality and the concept of interdependence centrality (ego network) [159]. Ego-net, a software program, gathers and analyses data from the egocentric social network. It assists the user in collecting and analysing all social network data from an online resource. It offers broad measurements of the whole network and data matrices that other applications may utilise for additional analysis. The Ego-net is a consequence of links that a particular Internet address sends and receives, and it is committed to gathering information about them and making them accessible to users in a meaningful way [185].

Regarding the topology, as depicted in Figure 2.6, *User1* requests content from the server at time  $t = 0$ , as all cache stores are currently empty. The content is being routed through:

$$CR_1 \rightarrow CR_2 \rightarrow CR_3 \rightarrow CR_4$$

From the server to User1. This mechanism ensures that only one of these routers caches the content. Assuming that User2 requests the same exact content, CR3 will satisfy the request. We can verify the process using the betweenness centrality, which shows that CR3 has the highest centrality value and the most content delivery paths passing through it. The main system for managing the network probably figures out each node's betweenness centrality offline first, as shown in Figure 2.6 [159].

By using centrality-based measures, the router with the largest value of betweenness will keep a copy (Centrality) [72]. The results demonstrated that the centrality-based mechanism could achieve a greater caching gain in real-time, regardless of the topologies on the synthetic network. The research of Chai et al. [159] proposed a centrality-based mechanism to improve cache performance and optimise server and cache hit rates. Researchers discovered that centrality works badly in many situations. The researchers hypothesise that the rigidly hierarchical structure of the Chinese Internet selects a relatively small number of routers near the core with high betweenness for caching [72].

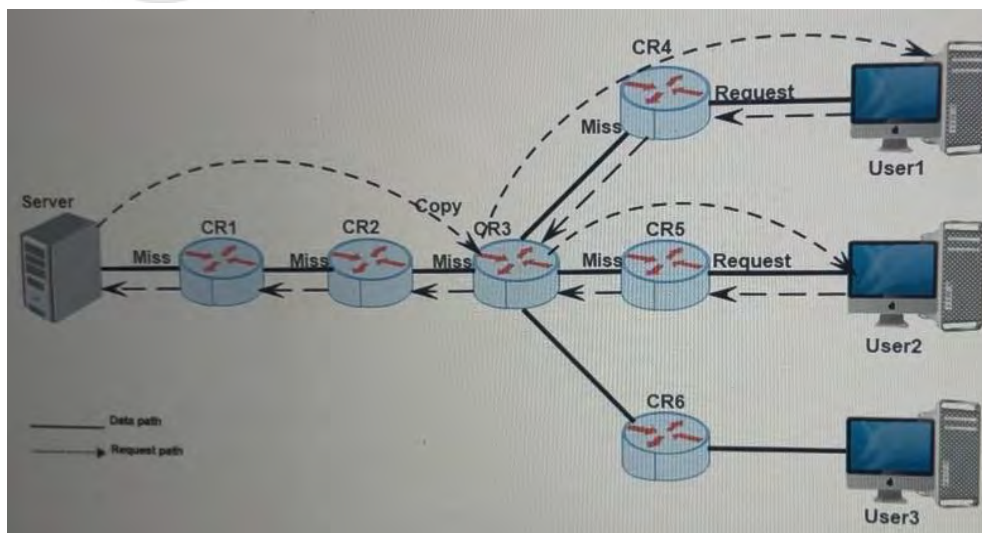


Figure 2.6. The Operation of Centrality-Based Mechanism

In VoD services, the node with the highest betweenness centrality only caches packets

or processed data once. This is the node with the fewest walkways. This approach maximises the benefits of betweenness centrality while preserving the material inside the node closest to the source. When used with dynamic systems, this mechanism makes it challenging to grasp the node location depending on the betweenness centrality in the ego network [159]. Furthermore, the centrality mechanism performs the worst in terms of delay performance when compared to LCD, LCE, Prob, PProb, Rand, and Cross. The possibility of a content node drop arises when various content delivery pathways link the network's cache misses [174].

#### **2.4.8 A Cache Placement Based on Compound Popularity Mechanism (CBCP)**

It is an implicitly coordinated mechanism that is based on two factors: the popularity of the content and the popularity of the node. By concurrently considering the content popularity and the node popularity over time, CBCP increases the reuse rate of data packets. Meanwhile, it can use a quick method to boost the content's popularity. According to a performance assessment in the Icarus simulator, the suggested mechanism outperforms various current mechanisms regarding cache hit ratio, start delay, and link load [67].

However, the findings demonstrate that the CBCP mechanism performs much better regarding start latency, cache hit ratio, and link load than the other mechanisms (LCE, LCD, ProbCache, and CL4M). Cache Less for More, also known as CL4M, is a Centrality-based caching mechanism, as was previously discussed in Subsection 2.4.7.

The concept of (ego network) betweenness centrality eliminates the performance uncertainty of the fundamental random caching approach and improves caching gain. Employing the idea of (ego network) betweenness centrality eliminates the performance uncertainty of the whole random caching approach, thereby accelerating caching growth [159]. Using the tree topology as indicated in Figure 2.7, the CBCP

assessment achieves about 0.02 better than LCD in a cache hit ratio.

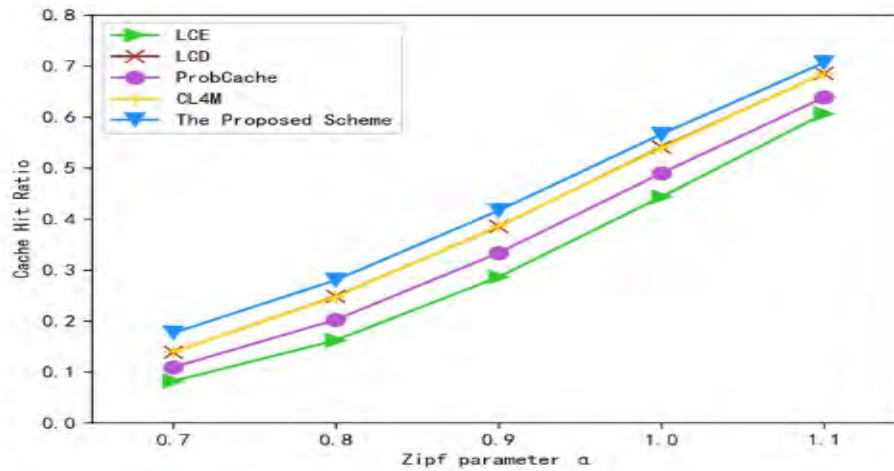


Figure 2.7. CBCP Evaluation [67]

#### 2.4.9 A Cache Placement Mechanism Based on Entropy Weighting Method and TOPSIS (CBEWT)

CBCP only includes a cache placement mechanism and also requires a small amount of communication overhead between nodes to obtain global data. Thus, CBEWT is designed to make more reasonable use of limited cache resources and avoid the communication overhead of obtaining global request information. Whereas CBEWT operates as a form of coordinated caching mechanism, primarily through implicit coordination between network nodes. The decision-making process is distributed but follows coordinated rules based on the entropy-weighted criteria and the TOPSIS ranking. CBEWT includes a cache placement mechanism and a cache replacement mechanism. The entropy weighting approach and TOPSIS simulate the consumer's request process to choose the optimal cache node for the cache item based on the node's current state. To decrease cache redundancy on the delivery line and increase the use of data packets in the nodes, it also includes two cache replacement mechanisms (an active cache replacement mechanism and a passive cache replacement mechanism). According to the various types of cache nodes, an efficient cache mechanism and data

packet migration mechanism further enhance cache performance [138].

The active cache replacement mechanism periodically clears redundant content items from the transmission stream to lessen cache redundancy. In contrast, the passive cache replacement mechanism intends to increase the node's utilisation of data packets. The findings demonstrate that CBEWT performs better regarding cache hit rate and latency compared to the other five caching means—LCD, LCE, ProbCache, CL4M, and CBCP. The link load performance of CBEWT is slightly less than that of the CBCP mechanism [67]. This cache placement mechanism registered around 0.001 better than the CBCP placement mechanism when  $\alpha = 0.8$ , as seen in Figure 2.8. CBEWT mechanism improves bandwidth efficiency, reduces latency, enhances load balancing, and dynamically adjusts to changing network conditions. It is ideal for delivering high-quality VoD services at scale. The implicit coordination between nodes ensures that the system functions efficiently without the need for direct communication, further supporting scalability and resource optimisation.

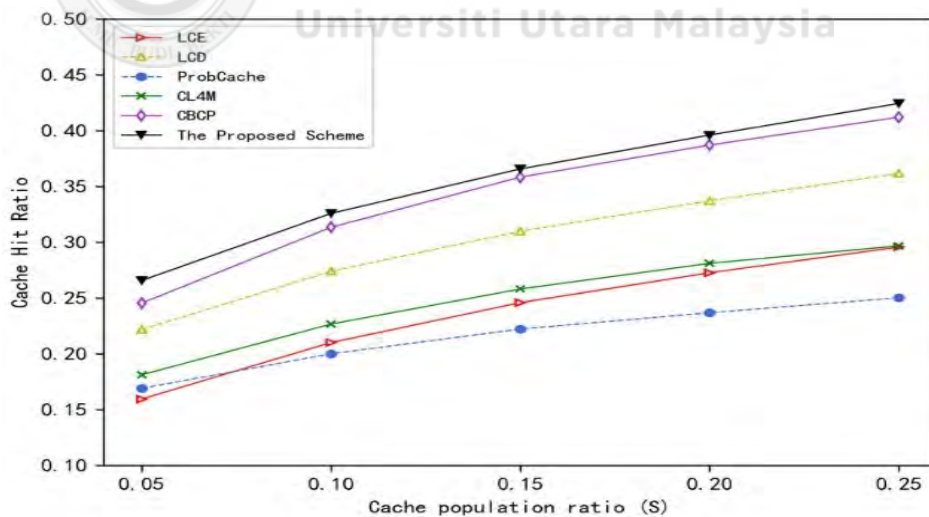


Figure 2.8. Evaluation of a Cache Placement Mechanism Based on Entropy Weighting Method and TOPSIS [138].

#### **2.4.10 Coordination Caching Placement for VoD in NDN**

The surge in VoD consumption, driven by user demands for low-latency, high-quality streaming, presents a unique challenge to network infrastructures. In NDN, which focuses on content-based communication rather than host-based routing, in-network caching emerges as a key enabler for efficient VoD delivery.

However, the mere existence of caches is insufficient. To meet VoD's stringent requirements—such as high Quality of Experience (QoE), minimal startup delays, and efficient bandwidth utilisation—there must be intelligent coordination of cache placement. This chapter critically evaluates existing cache placement strategies to introduce the novel explicit cooperative caching placement mechanism as a robust solution tailored to VoD demands because cooperative caching achieves efficient content distribution, reduces latency, and enhances the user viewing experience in bandwidth-intensive applications [187, 188, 189]. Furthermore, it ensures the maintenance of high-quality video streaming experiences [190]. The cooperated caching placement significantly reduces latency and buffering, thereby enhancing the QoE for end-users [189]. This approach improves cache hit rates and content retrieval speeds [191].

**VoD-Specific Requirements in NDN:** VoD content has unique characteristics, which are: large file sizes of HD/4K videos that consume significant cache space, content demand spikes unpredictably, content must be fetched in ordered chunks for smooth playback, and QoE sensitivity. Users are intolerant of delays, buffering, or resolution drops. Consequently, coordinated caching mechanisms must address content popularity dynamics, user behaviour, network topology, and resource constraints in an integrated manner.

Despite innovations, current mechanisms often fall short on key VoD requirements.

**Table 2.1**  
Coordination Caching Placement for VoD in NDN

Caching Placement Mechanism	Coordination Type	Strengths	limitations	Suitable for VoD
LCE [170]	None coordinated	High initial availability, simple	Excessive redundancy, poor cache diversity.	Too naive for VoD, leads to cache pollution and inefficient memory use.
LCD [163]	Implicit coordinated	Reduces duplication, improves distribution comparing to LCE.	Passive and path-dependent, lacks popularity awareness or temporal adaptation.	Not suited to the fluid demands of VoD traffic.
RAND [156]	Non_coordinated	Minimal overhead.	Unreliable, no content awareness. Poor hit ratios, especially for VoD.	Inefficient and unstructured, incompatible with VoD requirements.
Prob [173]	Implicit coordinated	Simple, less redundancy.	Random placement leads to cache inefficiency. Ignores content demand patterns	Prob reduces waste, but lack of content sensitivity limits VoD efficiency.
PProb [16]	Implicit coordinated	improves basic caching efficiency of Prob.	Falls short in addressing the temporal, behavioral, and quality-driven complexities of VoD traffic in NDN.	Improves over Prob, but doesn't handle trend shifts well. It lacks coordination needed for modern streaming applications.
Cross [100]	Implicit coordinated	Content-aware, supports user behavior modeling.	Complex to deploy, heavy communication overhead.	Strong potential for VoD, but may be too heavy for dynamic, large-scale deployment.
Centrality [159]	Explicitly coordinated	Optimizes content reach.	Ignores dynamic user demands, central nodes can become congestion hotspots.	Works well for generic content distribution, but not responsive enough for VoD traffic variability.
CBCP [67]	Implicit coordinated	Reduces redundancy; enhances content reachability.	Lacks predictive modeling, does not integrate viewer engagement metrics.	Better coordination, but lacks fine-grained VoD behavior analysis.
CBEWT [138]	Implicit coordinated	Adapts to content demand over time.	Can be sensitive to parameter tuning. Reactive, not predictive.	Promising for adaptive caching, but too reactive and parameter-sensitive for VoD.

However, most predictive intelligence systems are reactive and do not anticipate content spikes. QoE-Centric Design User experience metrics like delay and buffering are underused. Fine-Grained Viewership Modelling Engagement metrics (watch time, skip rate) are rarely considered. Cross-Node Coordination. Many schemes are node-local or weakly coordinated. These schemes often lack scalability and adaptability. Existing mechanisms struggle to scale in dynamic networks. The new proposed mechanism aims to address these critical shortcomings. It combines predictive analytics and coordinated caching to deliver VoD content efficiently and responsively. As VoD becomes the dominant form of content consumption, coordinated cache placement mechanisms must evolve to meet its complex and dynamic demands. Existing mechanisms offer useful foundations, but they fall short of predictive capabilities, user-centric modelling, and coordinated caching strategies. New cooperative caching placement addresses these gaps by unifying predictive analytics, behavioural modelling, and internode coordination to deliver a QoE-optimised, scalable, and intelligent caching framework for VoD in NDN. Its introduction marks a step toward a next-generation content delivery infrastructure, where user behaviour drives the network, not the other way around.

In NDN during VoD services, caching mechanisms play a crucial role in efficiently storing and retrieving content. Coordination refers to the degree of synchronisation or collaboration between nodes when placing content in the cache. Each caching mechanism exhibits different levels of coordination, which significantly impacts performance metrics such as bandwidth utilisation, latency, cache efficiency, and scalability.

For VoD services, where high demand and dynamic content popularity are common, mechanisms with explicit or implicit coordination offer the best balance between scalability, performance, and resource utilisation.

## 2.5 Comparison of Caching Placement Mechanisms

The Table 2.2 compares the mechanisms above as a short way to analyse and discuss the details of these mechanisms, how they work, and their advantages and disadvantages. This comparison is based on various aspects: stability, delay, the process of multiple nodes, feasibility in multi-platforms, advantages, and the complexity of each mechanism. Prob exhibits the highest level of stability, while Cross also demonstrates stability. Rand is highly stable in peer-to-peer network activities. Most network settings indicate that PProb is stable, whereas LCD can be considered stable. LCE only exhibits stability when processing packets in a single order. Finally, Centrality is not stable in most networks because it may become a bottleneck and lose data packets. Generally, the mechanism becomes simpler and more stable when it relies on fewer factors. While it is complex, it is not stable.

The table indicates that mechanisms like Cross and Centrality experience significant delays as the size of the data content increases. For instance, videos, which represent the large-size data, can contribute to delays. Specifically, LCD may experience delays when it fetches packets from third-party networks. Whereas third-party networks refer to these networks that are typically managed, owned, or operated by an external entity (the third party) and are distinct from the organisation's own IT systems or networks, such as Content-Centric Networks (CCN), companies like Amazon Web Services (AWS), and payment processing services like PayPal. Similarly, the Rand placement mechanism experiences delays due to multi-layer packet processing. Prob mechanisms, on the other hand, face delays because of inconsistent data processing. Each of these placement mechanisms provides various levels of delay performance based on the different reasons for the delay.

**Table 2.2**  
Summary Comparison of Caching Placement Mechanisms

Aspect Mechanism	Stability	Delay	Process Multiple Node	Feasibility in Multi-Platforms	Advantage
LCE	Stable when requesting a single packet	Getting delayed when requesting from a third-party network	Able to process multiple nodes within the same network	Low feasibility when processing packets on multiple platforms	Perfect selection for flash flood incidents
LCD	Stable in most network scenarios	Delay may occur by increasing cache size	Able to process multiple Ethernet nodes	Partly able to process packets in various platforms	Works better under workloads and interconnection
Rand	Highly stable in a peer-to-peer network	Some delays are due to multi-layer processing nodes	Unable to process multiple nodes to get the original packet	Get problems when processing packets on different platforms	Good when identifying the individual resources
Prob	Most stable in distributed networks	Delay may result when processing inconsistent data	Capable of processing multiple nodes	Deal with the high packet loads on multiple platform	Small hop distance, low redundancy, and less delivery time
PProb	Stable in most network settings	Delay can occur when processing multimedia sources	Allow multiple nodes to be stored	Support multiple platforms with a constant packet for each node's reference sequence	Reduces the number of miss occurs when items are requested
Cross	Stable in a distributed network	More delay when the items have a large memory	Support Multiple nodes	Have specific difficulties checking the nodes' references	An equal cache sizes of available variety routing methods
Centrality	Not stable in most networks	A certain delay amount when cache size increases	Support multiple nodes	Support a variety of platforms	Keeping the material in the node nearer the source

The LCE has the capability to process multiple nodes within the same network. The LCD is capable of processing multiple nodes over the Ethernet. The Cross, Centrality, Prob, and PProb mechanisms support the multiple nodes. At the same time, Rand cannot process multiple nodes when the reference is given to the original packet. On the other hand, the feasibility of LCE is limited when it comes to processing packets

across multiple platforms. In contrast, the LCD's feasibility is partially compatible. Centrality supports multiple platforms, whereas PProb supports multiple platforms, and the reference sequence for each node is consistent with the packet. Rand and Cross may encounter certain challenges when operating on multiple platforms.

Finally, the Prob placement mechanism provides the best advantages because it provides a short distance between the user and server, low redundancy, and less delivery time. Simultaneously, the other placement method offers a distinct advantage. In other words, LCE is an excellent selection for flash flood incidents. LCD works better with interconnection topologies and multiple workloads. The Rand mechanism has the advantage of supporting many platforms and identifying the order of individual resources along a network path. PProb reduces the frequency of misses during item requests. For several routing systems, it is possible to have cross-cache banks of the same size. The benefit of Centrality is the ability to store content inside a node closer to the source.

## **2.6 Summary**

This work presents and explains the general architecture of NDN, with the previous and current works in NDN and cache placement mechanisms. In addition, the contemporary mechanisms in NDN, dealing with the stream of videos during the consumer's requests, are reviewed. The recent cache placement mechanisms are debated, such as LCE, LCD, Rand, Prob, PProb, Cross, Centrality, CBCP, and CBEWT. Thus, the NDN needs to be improved by finding the features of the nine cache placement mechanisms in terms of delay time and by discussing them based on stability, processing multiple nodes, multi-platform feasibility, and complexity. Furthermore, the main advantage of these mechanisms is elaborated, with the finding that the Probabilistic placement mechanism is the best in delivery time. By combining

the analysis of earlier studies in NDN, VoD, and QoS, the proposed mechanism would positively impact Internet video viewers and publishers, considerably developing academic learning.

NDN is classified into three main aspects. The first is the NDN system architecture, which comprises PIT, FIB, and CS. The second aspect is NDN applications, which cover several fields and areas. Although the application area that this proposal focuses on is VoD, as explained. The third major aspect deals with NDN system services. These system services include Routing, Forwarding, Security issues, Mobility, and Caching. Note that caching has two types of mechanisms: the placement caching mechanism and the replacement caching mechanism. These mechanisms are based on various concepts to produce numerous other mechanisms. However, this proposal adopts a caching Placement mechanism based on the Path Explicit Coordination scheme and a dynamic partitioning scheme to introduce a new caching mechanism in NDN using VoD. In addition, this proposal will enhance the forwarding mechanism, which can achieve less delay in performance and enhance the QoS, which supports QoE and multiple environmental conditions. At the same time, the major aim of the dynamic partitioning scheme is to minimise the cache miss probability and ensure fairness across various NDN traffic. It is best suited for video, especially VoD.

## **CHAPTER THREE**

### **RESEARCH METHODOLOGY**

The methodology used to devise the proposed Named Data Networks (NDN) caching coordination placement mechanism for partitioned VoD is comprehensively detailed in this chapter. Accordingly, it can be inferred that the goal is to outline the mechanisms employed in carrying out the study and show how they might effectively achieve the study's goals. The aim of this research is to design and evaluate a new coordinated caching placement mechanism for partitioned VoD in NDN. Therefore, the main goals of validation, verification, and performance assessment of the new placement caching mechanism are covered in this chapter. Accomplishing this task requires adopting a particular methodology, so the research adopts the Question-Step-Objective Diagram (QSOD) by outlining each stage of the approach in detail.

To achieve the aforementioned aim of this chapter, several sections have been outlined. A detail of the research operation framework is presented in Section 3.1, to outline the research steps and the options of research methodology available. Then Section 3.2 will cover how to actualise the conceptual model, current study stages, and the proposed mechanism design. The next Section 3.3 is for verification and validation modules. The Section 3.4 of evaluation by achieving the simulation construction until arriving at the confidence level of simulation results. Utilising certain well-known NDN measurements, like average hit ratio, average End-to-End Delay (E2EDelay), server load reduction, and overall network footprint reduction, the new caching placement mechanism employing VoD in NDN will be tested. Finally, a summary of this chapter can be found in Section 3.5.

### **3.1 Research Operational Framework**

Defining the research problem is crucial to justify conducting the study. This problem determines the research subject and its aims and objectives. It also indicates the type of data needed for investigating the arising issues and the suitable type of analysis capable of enabling the researcher to arrive at conclusions which answer the research questions. Irrespective of the differences in research projects, in terms of subject matters, size or complexity, the process described is followed for practically all research projects. Research differs in that some projects create new knowledge, while others test and refine existing knowledge [192]:

A framework will be developed for the actual research motivation based on the answers to the research questions. Although the answers to these questions aren't easily obtained, this chapter delves into what's needed to provide suitable responses. This will be described in the form of a sequence of steps, which is more organised in comparison to the series of tasks that take place in reality. As more knowledge is gained during the research, it is important to present a diagram suitable to describe the major steps involved in the various phases and stages of the research work, and this is useful for planning out a work programme in the form of a timetable. The progress of the research work can be monitored through a comparison of the present stage of work with the steps in the process.

In the work of Birks et al. [193], it was stated that the research framework might also be thought of as the mechanism or layout that directs data collecting and processing. Researchers have emphasised that the research framework guides and structures the evidence compilation process. It also reflects guidelines set for the study's scope [193, 194]. Salter et al. [195] likewise stated that the primary reason why a research framework is required is to guarantee the collected evidence or information that is

required for the researcher to provide answers to the defined research question(s).

Previous studies have confirmed that the research framework has several stages representing all of the study's activities. In contrast, every phase includes several stages. This study has three phases of specific activities to provide the anticipated results after the tasks and activities, as illustrated in Figure 3.1. Each phase contains one question of this research, its objective, and the steps that must be performed to achieve this objective.

To achieve the research objective, the researcher needs to take certain actions [196, 68]. The figure shows these research-related steps 3.1. To meet the research goals, the steps have been divided into three phases. The first phase is to fulfil the first objective for the second and third objectives. Therefore, the following research steps were achieved based on the questions to satisfy the objectives of this research by proposing a new placement mechanism to enhance Quality of Service (QoS), especially the VoD services during NDN.

**Phase One:** This phase involves many steps, and these steps are summarised as follows:

1. **Background of study:** To perform an in-depth study that relates to the comprehensive literature analysis, leading to the definition of the research problems within the context of coordinated caching placement for partitioned VoD in NDN. This step underscores the challenges faced by the NDN caching placement mechanisms, especially in VoD environments. It offers essential background details and outlines the research's broader context.
2. **Literature review:** To analyse prior experimental research on NDN cache

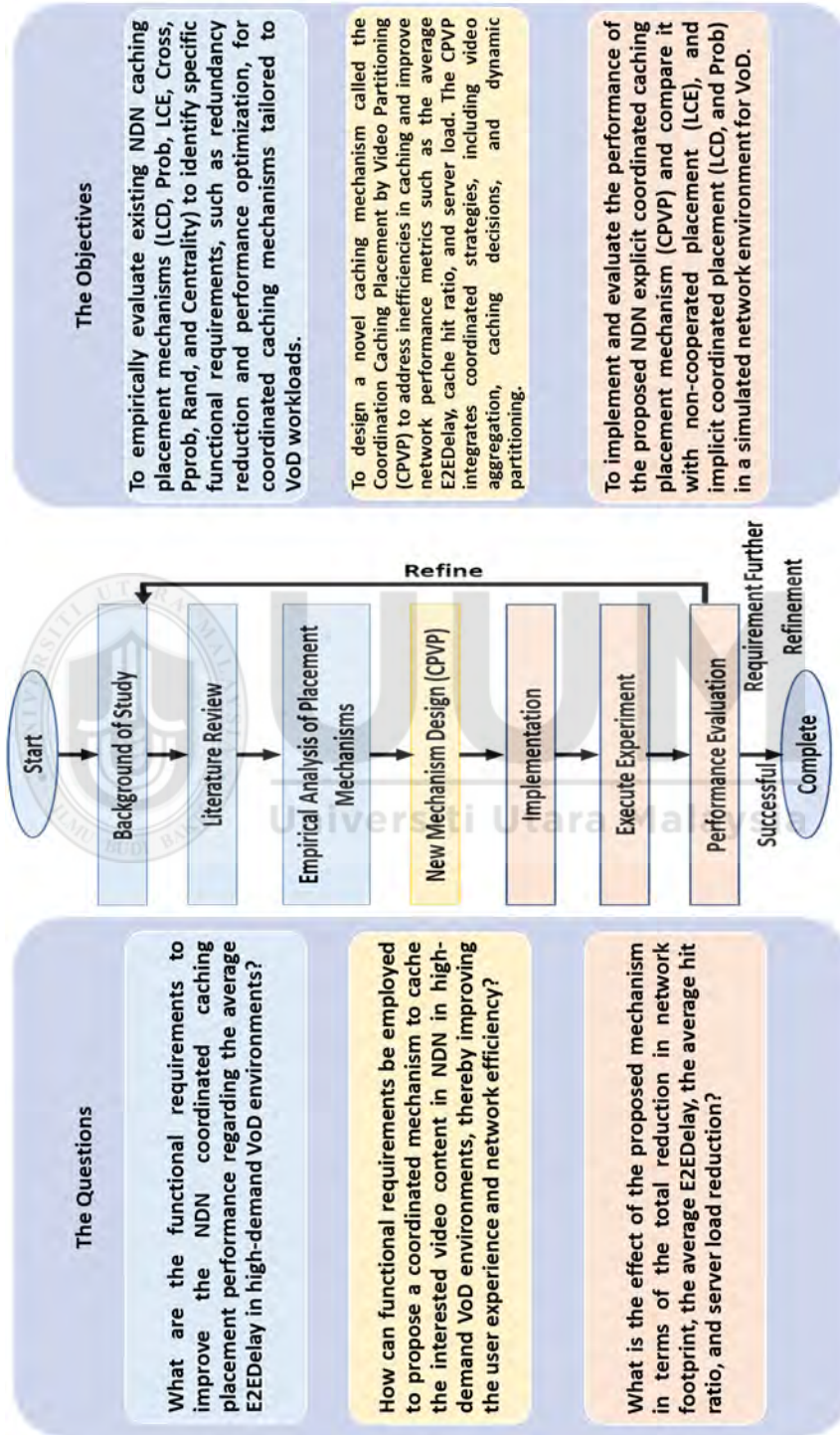


Figure 3.1. The Questions-Steps-Objectives Diagram

mechanisms specifically for partitioned VoD, identifying the causes for performance lags. The literature pertinent to this research's theme of coordinated caching placement for partitioned VoD in NDN is thoroughly and critically scrutinized.

3. **Empirical analyzes of placement mechanisms:** To execute an empirical evaluation to analyse and evaluate extensively the features of the famous placement mechanisms, which include LCE, LCD, Cross, Prob, PProb, Centrality, and Rand mechanisms. Similarly, using important performance indicators, determine the regions that should be improved or deleted for NDN cache placement mechanisms during VoD workloads: average hit ratio, average E2EDelay, total reduction in the network footprint, and server load reduction. After learning about potential improvements to the fundamental mechanisms, this stage produces a list of future enhancements to the original placement mechanisms in NDN during VoD. This step is achieved by executing the following:

- a) Implementing the seven placement mechanisms utilised in VoD workloads to determine the findings from the performance metrics as were previously mentioned by using Network Simulator ndnSIM2.0.
- b) Evaluation and analysis of the obtained results using experimental work and the parameters of video quality information, then comparing the results of these mechanisms and writing the report.

**Phase Two (New Design of Caching Placement Mechanism):** Designing a novel caching placement mechanism termed "Coordination Placement by Video Partitioning (CPVP)" for use in NDN with VoD services. The design process encompasses the establishment of design requirements, specifications, justifications, and objectives for the new mechanism. The goal is to reduce the

average E2EDelay and to enhance the average cache hit ratio, thereby achieving a significant reduction in network footprint and server load.

**Phase Three:** This is the pivotal phase of this proposal, concentrating on the implementation, validation, verification, and evaluation of the CPVP mechanism. The subsequent decision to refine or retain the new mechanism will hinge on the outcomes of this phase, which entails:

1. Implementing, validating, and verifying the design of the new mechanisms. The mechanism is set to be executed using the network simulator ndnSIM 2.0. Subsequent to this implementation, the design experiments will be established, with parameters chosen judiciously.
2. Conducting experiments across diverse scenarios. The outcomes from this phase will yield a dataset for performance assessment.
3. Undertaking a performance evaluation of the new mechanisms. This crucial step allows researchers to consider various design alternatives and pinpoint the most efficacious mechanism [196, 197].

For Figure 3.1, the thesis goes into great detail on each of the phases above. While the second step is thoroughly covered in **Chapter 2**, the first step has been expounded on in **Chapter 1**. The third step is completed in **Chapter 4**, while the fourth step of the new mechanism design stages and putting the new mechanism into use are explored in **Chapter 5**. Finally, in **Chapter 6**, an explanation of how the experiments, design, and findings were verified and validated. This section includes a description of the metrics used to assess the performance of the new tool and the particulars of the experiment setup.

### 3.2 The Proposed Conceptual Model of Coordination Placement of Video Partitioning mechanism (CPVP)

The NDN packet, which carries the request of the user for the desired content, is known as an 'interest', while the chunk having the desired content is known as a 'data chunk' [82, 83]. Despite the advanced caching mechanisms in NDN, inefficiencies persist in managing VoD traffic, particularly under high workload conditions. This study addresses these challenges by emphasising the need for coordinated caching algorithms that can efficiently handle simultaneous video requests while minimising average E2EDelay and server load. In the conceptual model, a module is proposed which is designed to support QoS by improving VoD services in NDN by minimising the delay performance in the network, as shown in Figure 3.2 by cooperating all the components cooperate, collectively achieving the goals of reduced latency, improved cache efficiency, and enhanced scalability.

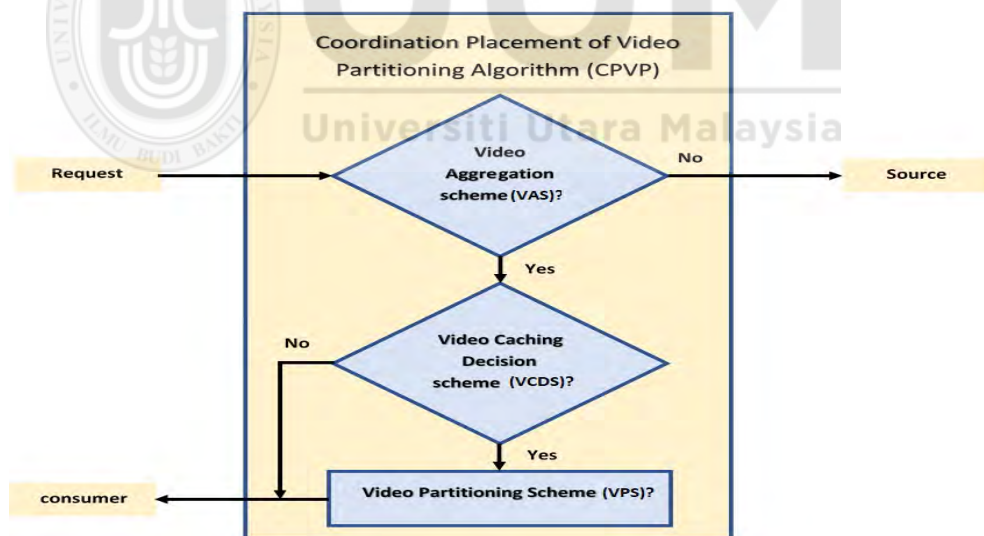


Figure 3.2. Conceptual Model of Coordination Placement of Video Partitioning Mechanism (CPVP)

This proposal discusses the design of a new cache placement mechanism called Coordination Placement of Video Partitioning Mechanism (CPVP). This mechanism caches the video after dividing it into three blocks dynamically. Each block is cached

on the edge router and its two neighbouring routers. This mechanism is based on explicit coordination on-path caching in NDN. It avoids congestion in network traffic by minimising repetition and caching content closest to the users. This mechanism is proposed to cache the chunks on the edge router and its two neighbour routers for a reduction in user-perceived delay, and a cache pipeline is progressively built on the path between the content origin server and the user. CPVP is achieved by combining three new schemes, which are VAS to deal with content retrieval efficiency, VCDS to optimise caching decisions, and VPS to enhance dynamic partitioning as follows:

1. **Video Aggregation Scheme (VAS):** It is designed to fetch the requested video content and find the distance from the consumer to the source. This video is found as a whole video on the server only. If the found video is only one block, then it needs to collect the other two blocks from the neighbouring routers.
2. **Video Cache Decision Scheme (VCDS):** This scheme is designed to decide whether to cache or not based on the location of the arrived router.
3. **Video Partitioning Scheme (VPS):** It is designed to partition the requested video into three blocks if the VCDS scheme is decided to cache. While the decision in the VCDS scheme is not cached, the data is backwards to the consumer.

Each of these schemes will be clarified more in Chapter 5. This research introduces a placement mechanism that does not rely on content popularity for caching decisions. The conventional approach of content popularity-based caching records and ranks content based on the number of historical requests it has received. Over time, cooperative cache placement can become inefficient, especially for video applications [187, 188, 189, 190]. The primary reason is that popularity-based mechanisms tend to allocate memory for content deemed popular, which might not always result in efficient

cache utilisation. Consider an extreme scenario where the same video file is requested by multiple users simultaneously. The chunks of that video, which were previously requested frequently, would be labelled as popular and thus would be cached for extended periods. This can lead to suboptimal cache use, resulting in increased delays in the network when newer, potentially more relevant, content is sought by users. Thus, popular content leads to the problem of an occupied cache, especially with large-scale content such as VoD.

The proposed placement mechanism, CPVP, addresses these challenges and offers a more efficient caching strategy, especially for video content. CPVP can dynamically partition videos and reduce network traffic by comparing with existing solutions like LCE, non-cooperative placement caching, and LCD, Prob, as implicit cooperative placement caching, while CPVP is considered as explicit cooperative caching. A detailed explanation of CPVP will be provided in Chapter 5.

### **3.3 Verification and Validation Model**

All relevant execution phases' verification and validation methods are implemented to ensure the execution is accurate. Ensure that the operational model accurately reflects the conceptual model and involves evaluating the simulation program [196, 68], Ghazali claims this is done to offer insightful responses to the questions under investigation [197]. Verification is regarded as the authentication that the numerical mechanisms actualised in a code can replace the original conceptual mechanism effectively, while validation is the affirmation that the code can give a precision forecast compared to the broadly recognised and substantial experimental information [198, 199]. This implies that verification of a model assesses the accuracy of changing the representation of a model from a pseudocode or flowchart form into a program executable on a computer [200, 201]. All mechanisms (models) must be validated to

ensure they have been correctly coded and are free of faults or flaws [202]. The CPVP mechanism has been verified. In Chapter 5, the specifics of the verification procedures are covered.

Models and simulations are commonly used to assist in the activities involved in Validation and Verification (V&V) through the life cycle of the system. V&V for any model ought to be done for every model use because the model may have the challenge of being valid for a certain class of conditions and invalid otherwise [203]. Verification of the model guarantees that the conceptual model is accurately programmed and its execution is reasonably stable [204]. The two renowned verification methods are checking the model and proving the theorem [205, 206].

The major processes for validating and verifying this research are shown in Figure 3.3. NdnSIM2.0, the network model, was first generated, after which the parameters and individual statistics were established. The seven mechanisms of caching placement will have their cache hit rates examined to validate the ndnSIM simulator. They include the LCE, LCD, Prob, Pprob, Cross, Rand, and Centrality mechanisms.

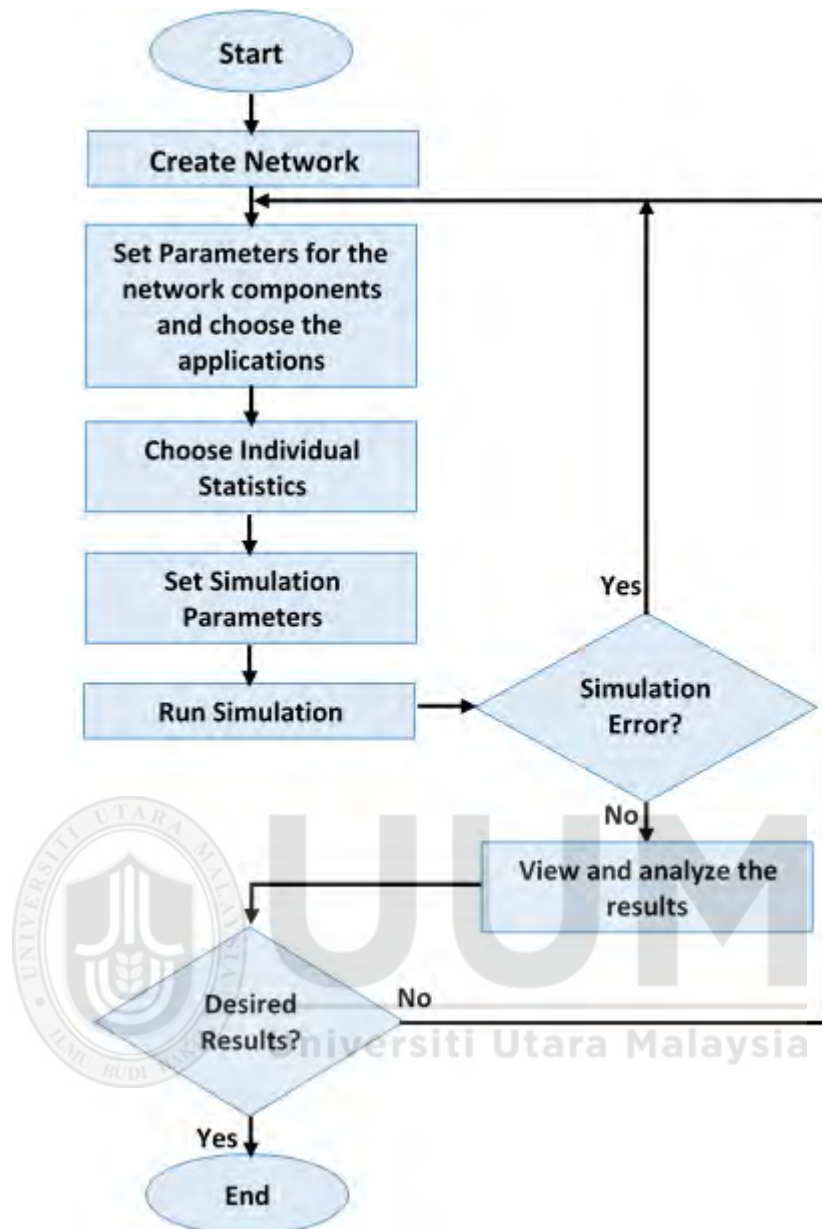


Figure 3.3. Main Steps for Validation and Verification

### 3.4 Evaluation

The performance evaluation of this research is achieved by adopting three possible conventional approaches [207, 208, 209], which include **analytical modelling, simulation, and measurement**. Selecting an evaluation approach is an important step in all performance evaluation projects [210]. Table 3.1 discusses the advantages and disadvantages of the performance evaluation methods that are mentioned in [211]. A set of equations developed using mathematical symbolism to represent how well a real

**Table 3.1**

Comparison of Performance Evaluation Mechanisms

Criteria	Analytical modeling	Simulation	Measurement
Time required	Low	Medium	High
Accuracy	Low	Moderate	High
Tools	Analysts	Computer languages	Instrumentation
Trade-off evaluation	Easy	Moderate	Difficult
Cost	Small	Medium	High

system operates is known as analytical modelling (**mathematical modelling**) [207]. Computer programming, which analyses operations utilising functional relationships in the system, can be used to examine mathematical models [211]. Performance evaluation based on measurement can be put into practice using test beds or by performing a real network, despite analytical modelling being saddled with the drawback of low accuracy compared to other performance evaluation methodologies. Similar to the previous method, this one is ideal due to its ability to deliver accurate findings. Still, it needs to catch up in terms of cost due to the particular kinds of equipment needed [212]. According to [213], it is very clear that **the simulation** would be the most desired approach for performance evaluation rather than analytical modelling and measurement, as shown in Table 3.1. Therefore, this research adopts the simulation approach as discussed in Subsection 3.4.1.

The evaluation approach requires evaluation criteria and performance metrics [214] discussed in 3.4.2.2. While Pentikousis et al. emphasised that, since ICN is still an emerging area, the community is still in the process of developing effective evaluation environments, including simulators, emulators, and testbeds [18]. Figure 3.4 exemplifies the three mechanisms mentioned above for assessing a communication network's performance [207]. In-network caching is a major characteristic of ICN, although there are still some issues that exist in NDN caching, such as how to place content replicas within cache nodes for maximisation of the cache system benefits without the introduction of excessive overhead. The enriched performance evaluation

is achieved by adding more practical topologies (see Subsection 3.4.1.3) to discover notable findings through in-depth analysis and comparison [215, 26].

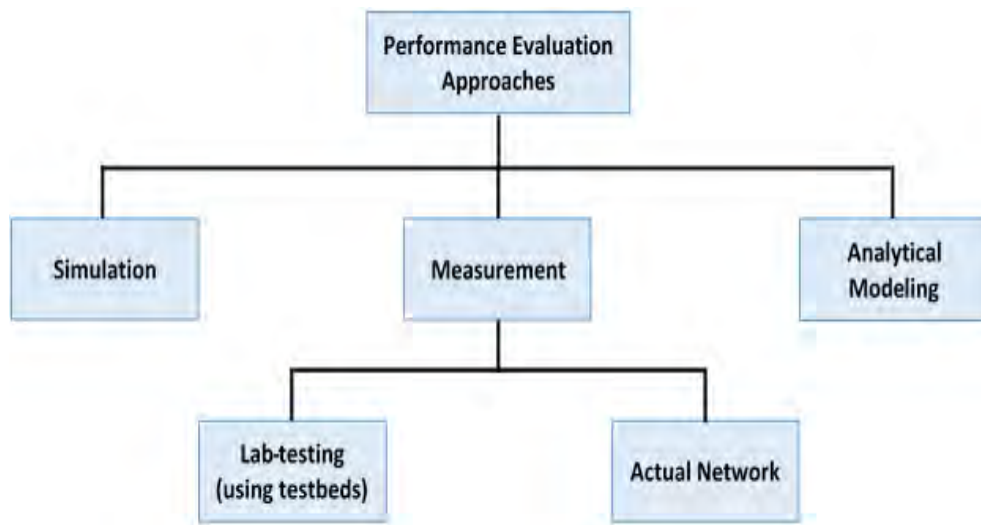


Figure 3.4. Performance Evaluation Approaches

### 3.4.1 Simulation Construction

Figure 3.5 illustrates the steps of the simulation structure of our research (the details are in Chapters 4 and 6). This structure contains seven steps, which are explained briefly as shown in the figure:

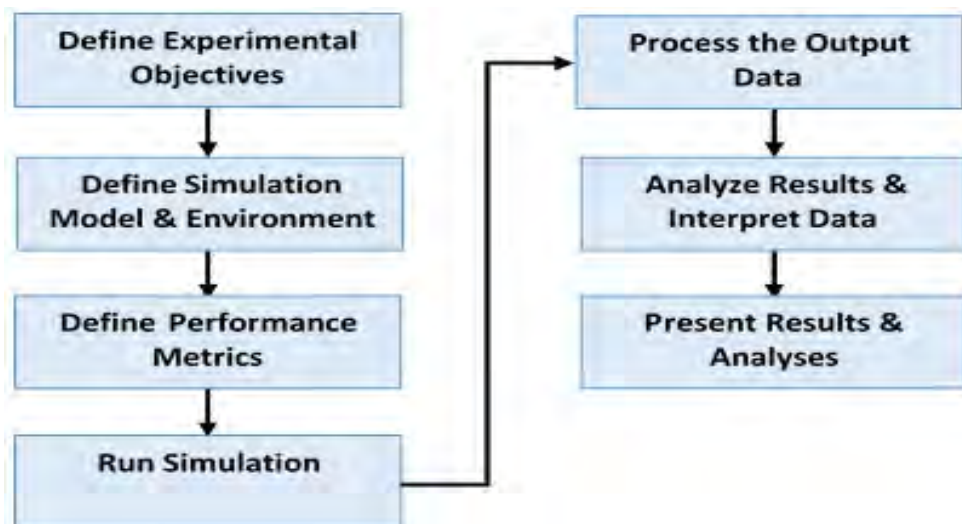


Figure 3.5. Simulation Construction

1. Define Experimental Objectives: In this step, the research objectives are planned

into a set of simulation experiments to achieve the system's evaluated goals and objectives, in addition to explicitly stating the evaluation criteria. The goal of this thesis is to implement and evaluate multiple caching placements in NDN during VoD services to get the required functionality. This requirement is used to design a novel caching placement mechanism termed "Coordination Placement by Video Partitioning (CPVP)" for use in NDN with VoD services.

2. Define Simulation Models and Environment: This step produces the analysis of the problem to be outlined and assigns a simulation model as in Subsection 3.4.1.1. The scenario, topology, and setting of simulation parameters are described in Subsections 3.4.1.2, 3.4.1.3 and 3.4.1.7, respectively, in this chapter. These parameters are required to design a novel caching placement mechanism, CPVP, for use in NDN with VoD services.
3. Define Performance Metrics: In this step, the performance metrics of the simulation are explained in detail. In this proposal, several performance metrics (such as the average hit ratio, the average E2EDelay time, total reduction in the network footprint, and server load reduction) are utilised. These metrics are explained in subsection 3.4.2.2 while a more detailed elaboration will be given in Chapters 4 and 6.
4. Run Simulation: The performance metrics, the simulation model, and the environment (topology, scenarios, and configuration) are gathered using network simulator (ndnSIM2.0) input scripts and run. This step is called a simulation phase. For our experiments in this research (Chapters 4 and 6), each simulation is run ten times to get the averaged results, quoted with error bars for confidence intervals of 95%.
5. Process the Output Data: The output data from the simulation phase will be analysed. This step is called the Post Simulation Process.
6. Analyse Results & Interpret Data: This phase involves the analysis of results, which were acquired from the simulation. It debates, evaluates and explains the

obtained results. This phase is called Results Analysis and Data Interpretation.

7. Present Results & Analyses: Presenting and elaborating the results from the simulation are required in this phase. This phase is crucial because it deals with the final achievement of the performance evaluation, which is the presentation of the results to give an insight into the system.

#### **3.4.1.1 Simulation Model**

The simulation approach is widely used in science, such as in engineering fields and computer science. Simulation refers to designing a virtual framework that simulates an actual one in real life [216, 217]. The fundamental concept of the NDN communication paradigm from IP principles needs an in-depth evaluation through experimentation, and simulation is an adequate approach to implement the testing at scale. The simulators that support the NDN environment are:

1. NS3: The open-source simulation that supports the NDN, whereas the address location of the NS3 code is (<http://www.nsnam.org/ns-3-22/>) [218].
2. ndnSIM: The first version of ndnSIM (NDN simulator) was introduced in June 2012 as an open-source NS3-based NDN simulator, and since that time, ndnSIM has evolved substantially, leading to the introduction of ndnSIM 2.0 [213].
3. ndnSIM2.0: The updated simulator of ndnSIM, whose first version was released in January 2015 [219, 220].

The goal of ndnSIM 2.0 is to match the simulation platform to up-to-date advancements in NDN research. For this reason, it adopts the use of the ndn-cxx library (NDN C++ library with eXperimental eXtensions) and the NDN Forwarding Daemon (NFD) to allow experiments with real code within a simulation environment [213].

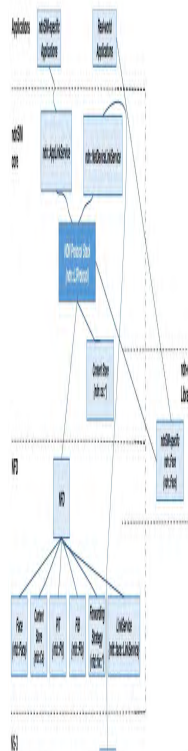
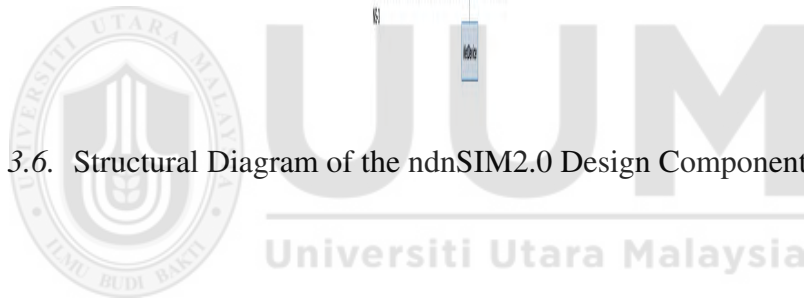


Figure 3.6. Structural Diagram of the ndnSIM2.0 Design Components [213]



Therefore, to simulate NDN, ndnSIM2.0 based on NS-3 will be best utilised, and Figure 3.6 shows the entire architecture of ndnSIM2.0. As shown below, the NFD module at the bottom of the architecture implements forwarding management, where PIT, FIB and CS data structures all belong to NFD. The ndnSIM2.0 core module initiates the implementation of the NDN protocol stacks, and the packet format of the ndnSIM2.0 protocol follows the ndn:L3Protocol specification [213, 221]. This new protocol is supported by every device in the simulation environment. The upper layer module is an application layer to the application code written in the ndn-cxx library [222]. These libraries are utilised for writing different applications for simulating the consumer-producer environment. ndnSIM2.0 also supports all NS-3-based functionality, and the cache replacement policies are also inherited in ndnSIM2.0. The simulator provides detailed packet tracing and manipulation-related

libraries, whose use can investigate traffic analysis, delay and performance analysis [223].

Given the nascent stage of NDN and the absence of a full-fledged implementation, choosing simulation as an evaluation instrument is both logical and fitting. Simulations provide a controlled environment where different parameters can be finely tuned and studied. In this environment, we can observe the mechanism's behaviour without the complexities and uncertainties inherent in live networks.

Although test beds provide a tangible platform for evaluation, creating a representative test bed for a system such as NDN presents significant challenges. Ensuring that such an environment is scalable and reflects real-world conditions can be demanding in terms of resources and technical feasibility. Furthermore, gauging the performance of a mechanism within a test bed, especially for an emerging technology like NDN, might introduce variables that are difficult to control or account for.

The primary objective of this research is to understand the performance of the mechanism under consideration. Simulation stands out as a more straightforward tool for achieving this objective. Using simulation, researchers can zoom in on the mechanism itself, effectively isolating it from extraneous variables that might be present in a more physical setup, such as a test bed. This approach ensures a more precision representation of its performance characteristics.

Given the current state of NDN and the complexities of establishing a representative test bed, the decision to use simulation as the main evaluation tool is both wise and appropriate for the research's scope.

### 3.4.1.2 Simulation Scenario

We will model the proposed work using the open-source, NS-3-based NDN simulator ndnSIM2.0, which became available in January 2015. We choose the LRU mechanism for replacement because it yields the best results, as reported in [224, 225, 226]. The Basic NDN Forwarding mechanism is used. Also, the evaluation will use four topologies: Tree, Abilene, GEANT, and Deutsche Telekom (DT), with each result obtained by averaging over ten simulation runs.

In this study, we have Chapter 4, and Chapter 6 is for evaluation. In Chapter 4, we compare the seven caching placements with each other. In Chapter 6, we use ndnSIM 2.0 in our simulations to test the new placement caching mechanisms, which are Coordinated Placement Video Partitioning (CPVP). We have selected LCE, LCD, and Prob as the current placement mechanisms for comparison with them. Whereas LCE is non-cooperated placement caching and (LCD, Prob) is implicit cooperated placement caching, CPVP is considered explicit cooperated caching. This will let us see how it works in comparison to other methods.

### 3.4.1.3 Network Topology

We can characterise the topological structure of a network, known as network topology [227, 228], either logically or physically. The graphical application theory [229] models communicating devices as nodes and the connections between them as links or lines. Chapter 2 evaluated the effectiveness of NDN caching mechanisms using various topologies. The 6-level binary tree with 127 nodes is the subject of Psaras et al.'s [16] analysis. The root node manages all requests, while the ISP's last two layers fulfil all of them. Li et al. [230] used a 4-level tree architecture, with each router's total content storage size varying from 10% to 50% of the overall content. Chai et al. [159] used a 4- to 6-level k-ary tree, with each node containing 2 to 5 offspring. Rossi et al.

[231] use five topologies: Abilene, Level 3, Tiger, DTelekom, and GEANT. Cho et al. [232] use the ISP-level topology of 1 transit domain and 5 stub domains. There are 5 routers in the transit domain and 10 in each stub domain. Additionally, 10 servers in the architecture handle requests from 1,000 connected users. Bernardini et al. [233] employed four ISP-level topologies: DTelekom, GEANT, Tiger, and Abilene.

However, there is no widespread agreement on the optimal topology for a fair evaluation of caching mechanisms. Because topology directly affects the outcomes of simulations, it is a crucial topic [231]. Thus, we evaluate the CPVP mechanism using four topologies: Tree [234, 235], Abilene Network Topologies [236], GEANT [237], and DTelekom [238]. To ensure fairness in the results, we equalise the number of routers for each topology to 127, add a single server, and limit the requests to the last two neighbour edges. However, the tree topology employs a 6-level binary system, comprising a total of 127 nodes. The root node represents the server node, and the requests come from the last two levels of the tree, assuming that individual users are not connected to backbone network routers; we note, however, that results are similar in the case of users attached to all routers.

## **1. Tree Network Topology**

The Tree network topology is a network structure which takes the form of a multipoint electrical circuit, with multiple branches of the central trunk, or root, bus. The tree network topology is a variation of the bus topology, being a hybrid network topology in which star networks are interconnected via bus networks, with the provision of just a single path between any two nodes. Tree networks are hierarchical, and each node has the ability to possess an arbitrary number of child nodes [239]. The works of [239, 57, 240, 241] show the application of this network topology.

## **2. Abilene Network Topology**

It has an impact on almost every significant innovation that defines our modern digital lives while continuing to define “what’s next”. It is also called Internet2. Internet2 controls the largest growing coast-to-coast research and education community that was built to deliver advanced, customised services which can be accessed and secured with the aid of the community-developed trust and identity framework. Internet2 services cut across all sectors of the nation, including 331 US universities, 60 corporations of government, and 43 regional and state education networks, through which support is provided for over 94,000 community anchor institutions, more than 900 incoming participants, 65 major companies working with the community, and 61 national research and education network partners that constitute more than a hundred nations [242].

Abilene is an advanced network, which was built by and for the education and research community, with the Internet2 network offering 100GB Ethernet technology on its entire footprint. By providing uninhibited performance on a profoundly programmable platform, the Internet2 network propels forward education and research. The network further provides transparent operations with continuous evaluation and optimisation by the community to satisfy the unique evolving needs of high-speed collaboration and research [242]. These research [243, 244, 245, 246, 247] are examples of studies adopting this topology.

## **3. GEANT Network Topology**

The GEANT network topology interconnects Europe’s National Research & Education Networks (NRENs), providing research data communication across the continent. The network is a Pan-European communications infrastructure with commendable

features of being the fastest and most reliable to serve Europe's education and research community. Being co-funded by European NRENs and the EC, the GEANT network project (coined GN3) is advancing into its third generation, together with associated development activities. The major objective of GEANT is the delivery of real value and advantage to society through the enabling of research bodies across Europe and the world at large to transform the approach of collaboration on groundbreaking research [248]. GEANT network topology has been utilised and discussed by various researchers [249, 106, 57, 60, 250].

#### **4. Deutsche Telekom Topology**

Deutsche Telekom (DT), which by revenue, has hitched the title of being the largest telecommunications provider in Europe is a German telecommunications company formed in 1996 with its headquarters situated in Bonn. As of July 2018, the German government holds a 14.5% stake in company stock directly, and another 17.4% through the government bank KfW [238]. The company is also a component of the Euro Stoxx 50 stock market index [251].

Deutsche Telekom Stiftung was established in 2003 to strengthen Germany's position as the leader in education, research, and technology. With 150 million euros of capital, it ranks as one of the country's major corporate foundations. It is dedicated to improving education in the fields of Science, Technology, Engineering, and Mathematics (STEM), as well as in digital teaching and learning. The foundation's activities are focused on four key areas: Education Drivers, Education Opportunities, Education Innovations and Education Dialogue [238, 252]. An example of researchers using this topology can be found in [253, 254, 255, 106].

#### 3.4.1.4 Content Popularity

The content popularity model is a tool that tracks the worth of each content's popularity. It is a function that, in other words, counts the specified number for each element. According to the research, only a handful of websites out of millions garnered the most visitors [256]. As a result, the material's popularity may be predicted using a power law-shaped probability distribution function [257]. The power law distribution, often known as the Zipf distribution and its extended variant Mandelbrot-Zipf (MZipf), is frequently used in the majority of models and simulations [258, 231, 257]:

$$pmf(x, N, q, \alpha) = \frac{(q+x)^{-\alpha}}{\sum_{k=1}^N (q+k)^{-\alpha}} \quad (3.1)$$

When  $N$  is the number of elements,  $\alpha$  is the distribution's slope,  $q$  indicates its plateau, and  $pmf$  stands for the probability mass function. A flatter distribution is achieved when  $q$  tends to zero as in [257]. According to the literature study, the alpha parameter for Zipf or MZipf can be between 0.5 and 3.0. In a simulation based on the MZipf distribution, content is chosen as needed depending on its likelihood value. Depending on how the  $\alpha$  option is configured, some items may receive more requests than others.

Saino et al. [259] use a Zipf popularity model with alpha values between 0.6 and 1.1, consistent with a range of alpha values used in the literature. This parameter's value in [16] is 0.8. While Zhang et al. [178] and Li et al. [230] assess with 0.3 and 0.9 for alpha, respectively. This parameter's value in [159] is 1.0. Sung et al.'s evaluation uses an alpha range of 0.25 to 1.5, while that of [260] is between 0.6 and 1.2. The alpha is tested for 0.85 in [232]. While Ong et al. [261] use this number between 0.5

and 3.0, Bernardini et al. [233, 257, 262] utilise values between 0.65 and 2.0. The alpha in [178] is 0.7, whereas [263] shows a range of 0.7 to 0.96. Alpha in [231] falls between 0.6 and 2.5. The caches are evaluated by other studies using traces taken from the Akamai CDN Asia [257], such as Fayazbakhsh et al. [264]. There is no consensus on the value of the MZipf distribution's alpha parameter since it is thought to be the best candidate to reflect content popularity. The fact that this number may range from 0.5 to 3.0 suggests that it might have a significant impact on how well the caching mechanisms function. Therefore, this study of the CPVP mechanism considers  $\alpha$  as 0.65; additionally, our design is not dependent on popularity due to the huge size of the video and to avoid occupation. However, the performance metrics are affected positively  $\alpha$  because it refers to requests again and again with the same interest, as concluded from our simulation results in Chapter Four.

#### **3.4.1.5 Cache Size**

Each node's space available for caching content objects is referred to as the cache size. The cache size is expressed as an absolute value or a ratio to the catalogue size; for instance, 20 elements or 0.02 of the catalogue size [257] are equal. Cache size 0.2 was employed by Saino et al. [259]. The 0.1 cache size was utilised by Chai et al. [159] and Cho et al. [232]. Rosensweig et al. [265] employ a cache size ranging from 0.1 to 0.4. However, Li et al. [230] take the number between 0.1 and 0.5. This figure of 100 is used by Zhang et al. [178]. Utilising 100–1,000 element cache sizes are Kang and Bilal [260]. Sung et al. [266] take 2,500 pieces, with a 2GB element size per piece and a 1TB cache. Wang et al. [178] made the simulation scenarios using 50Mb and 200Mb cache capacity with 100Mb and 300Mb content size. At the same time, Ong et al. [261] take cache sizes from 0.05 to 0.3. At the same time, this evaluation of the CPVP mechanism uses cache sizes from 1GB to 1TB.

#### **3.4.1.6 The catalogue size**

It refers to the total number of contents in a network. For the catalogue size, a distinct range of values has been used. This value  $10^3$  is used by Zhang et al. [267] and Chai et al. [159]. Rossi et al. [231] based on a YouTube-like catalogue that contained  $10^8$  objects with an average 10MB chunk. However, Kang and Bilal and [260] used the catalogue size of  $10^4$ . Bernardini et al. [233] consider a catalogue of  $10^6$  elements in their experiments. While Ong et al. [261] consider  $10^4$  elements. Some academics have not included the specific catalogue size in their writing. In addition to the authors in [16, 259, 264, 164] just stated that they receive  $10^5$  queries for each catalogue. Thus,  $10^6$  content is used in this evaluation of the CPVP mechanism.

#### **3.4.1.7 Simulation Parameter Settings**

In this work, VoD is used in the evaluation of the caching placement mechanisms (LCE, LCD, and Prob) with the proposed mechanism (CPVP). While analysing the suggested mechanisms in the NDN, VoD would also be looked at for operational entries. The VoD traffic is obtained randomly from the Scalable Video Coding-Dynamic Adaptive Streaming HTTP (SVC-DASH) as explained in Subsection 3.4.2.1.

**Table 3.2**

Simulation Parameter Settings

<b>Topologies</b>	<b>Tree, Abilene, GEANT, and Deutsche Telekom Network Topologies.</b>
<b>Replacement mechanism</b>	<b>LRU.</b>
<b>Forwarding mechanism</b>	<b>Basic NDN Forwarding.</b>
<b>catalogue size</b>	<b>1000,000 contents.</b>
<b>Chunks block size</b>	<b>1/3 of file.</b>
<b>Zipf exponent</b>	<b><math>\alpha</math> parameter 0.65, and 2.0.</b>
<b>Cache size</b>	<b>1GB, 10GB, 100GB, and 1TB.</b>
<b>Number of Simulation</b>	<b>10 runs.</b>
<b>Simulator</b>	<b>ndnSIM2.0.</b>
<b>Evaluation metrics</b>	<b>Average Hit Ratio, AverageE2EDelay, server load reduction, and total reduction in the network footprint.</b>
<b>VoD traffic</b>	<b>The Scalable Video Coding-Dynamic Adaptive Streaming HTTP (SVC-DASH).</b>

Regarding the parameter settings, all that is needed by the simulation is considered, such as catalogue size, Content Popularity Model, topologies and cache size. In this case, a standard value is chosen for all the parameters. The scenarios also consist of a catalogue with 1,000,000 files of video. The cache size is also simulated in four cases of cache sizes, which are 1GB, 10GB, 100GB, and 1TB. Content Popularity Model uses a probability distribution function such as Zipf based on parameters 0.65 and 2.0. The parameter ranges largely from 0.6 to 2.5 [233], but for this experiment, it is considered as 0.65, which refers to a low popularity scenario to avoid bias for any popular content. LRU working based on page selection for a replacement has not been referenced for the longest time, but considered mostly used as in [159, 226]. The

various simulation parameters to be used in this research are listed in the table 3.2.

### 3.4.2 Simulation Setup

To evaluate the new mechanism, CPVP, the performance metrics results are compared with these placement mechanisms: LCE, LCD, and Prob, to confirm that the simulation results of CPVP are valid. The reason for the comparison with LCE, LCD, and Prob is that LCE is the default NDN caching mechanism; LCD is developed from LCE, while Prob is getting a good result in performance metrics in Chapter Four of this thesis. Additionally, the other reason is that each of these three mechanisms can be considered an example of one type of caching mentioned in Chapter Two. The LCE mechanism is non-coordination on path caching, as mentioned in Subsection 2.3.2.1 and explained in detail in Subsection 2.4.1. Whereas the LCD mechanism is an implicit cache coordination scheme, as mentioned in Subsection 2.4.2 and explained in detail in Subsection 2.4.2. Prob is also in the same type of LCD, but it efficiently decreases redundancy, and big minimises the E2EDelay performance metrics. Therefore, our design represents the explicit cache coordination on the path. For more precision, see Figure 2.3.

The evaluation of our implementation by cache size from 100 content elements to 100,000 elements (1GB to 1TB sequentially), a catalogue size of  $10^6$  elements, while a Zipf popularity equal to  $\alpha = 0.65$  (see Table 3.2). In the literature,  $10^4$  contents is the widely utilised catalogue size [268, 260, 261, 269], but because of the current traffic nature, it isn't easy to guess the Internet simply with  $10^4$  contents. As a result, in our studies, the catalogue size was set to 106 items, which is thought to be sufficient for assessing the performance of Internet traffic [233, 262]. Files are represented by the components and contents. As a result, they are both used equally in this thesis. In the simulation outcomes, the median values of ten runs are evaluated for Average Cache Hit, Average E2EDelay, Server Load Reduction, and Reduction in Network Footprint.

The basic NDN Forward mechanism is used according to what was considered the best forwarding mechanism by the study of Aloulou N. et al. [65]. Assuming that the CPVP is cached in the three blocks in the three neighbour routers at the same time, therefore, this study used Last Recently Used (LRU) as a caching replacement mechanism. However, each block is equal to  $1/3$  of a file. This study achieved simulation by using these topologies: Tree, Abilene, GEANT, and Deutsche Telekom Network Topologies in ndnSIM2.0. The VoD traffic is obtained randomly from the Scalable Video Coding-Dynamic Adaptive Streaming HTTP (SVC-DASH).

#### **3.4.2.1 The Model of Video**

This experiment sends random video packets based on the video content which are initially from the dataset of the Scalable Video Coding-Dynamic Adaptive Streaming HTTP (SVC-DASH). The open-source toolchain can be accessed from the GitHub repository at <http://tinyurl.com/DASHSVC> [270, 271] by using the dataset at [tinyurl.com/SVCDataset](http://tinyurl.com/SVCDataset); see Figure 3.7. In this figure, each 48 frames refer to a segment size (2 seconds at 24 fps). A dataset of DASH/SVC containing multiple videos at varying bitrates and including spatial 1080p resolutions. Second, a toolchain for multiplexing SVC encoded videos is provided, therefore making our results reproducible and allowing researchers to generate their own datasets [270, 271]. In DASH systems, each video consists of multiple segments or “chunks” (corresponding to a few seconds of playtime) and each chunk is encoded at multiple discrete bitrates [177].

### Source Videos

The following videos are part of our dataset:

Name and Source	Length	Variant I (360p/720p/1080p)	Variant II (480p/607/720p/1080p)	Variant III (spatial+quality)	Variant IV (spatial+quality)
<b>Short Movies from <a href="http://media.xiph.org">media.xiph.org</a></b>					
Big Buck Bunny YUV420p, 1080p, 24 fps	14315 frames 299 segments	360p@24fps: MPD, Segments 360p@6, 12, 24fps: MPD, Segments 720p@24fps: MPD, Segments 720p@6, 12, 24fps: MPD, Segments 1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments	PSNR/SSIM: 360p@24fps: MPD, Segments 360p@6, 12, 24fps: MPD, Segments 720p@24fps: MPD, Segments 720p@6, 12, 24fps: MPD, Segments 1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments	1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments PSNR/SSIM: BL, EL1, EL2, EL3	1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments PSNR/SSIM: BL, EL1, EL2, EL3, EL3
Elephants Dream YUV420p, 1080p, 24 fps	15691 frames 327 segments	360p@24fps: MPD, Segments 360p@6, 12, 24fps: MPD, Segments 720p@24fps: MPD, Segments 720p@6, 12, 24fps: MPD, Segments 1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments	PSNR/SSIM: 360p@24fps: MPD, Segments 360p@6, 12, 24fps: MPD, Segments 720p@24fps: MPD, Segments 720p@6, 12, 24fps: MPD, Segments 1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments	1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments PSNR/SSIM: BL, EL1, EL2, EL3	1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments PSNR/SSIM: BL, EL1, EL2, EL3, EL3
Sintel PNG, 1080p, 24 fps (converted to YUV420p)	21317 frames 444 segments	360p@24fps: MPD, Segments 360p@6, 12, 24fps: MPD, Segments 720p@24fps: MPD, Segments 720p@6, 12, 24fps: MPD, Segments 1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments	PSNR/SSIM: 360p@24fps: MPD, Segments 360p@6, 12, 24fps: MPD, Segments 720p@24fps: MPD, Segments 720p@6, 12, 24fps: MPD, Segments 1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments	1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments	1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments
Tears of Steel YUV420p, 4k, 24 fps (converted to 1080p)	17620 frames 368 segments	360p@24fps: MPD, Segments 360p@6, 12, 24fps: MPD, Segments 720p@24fps: MPD, Segments 720p@6, 12, 24fps: MPD, Segments 1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments	PSNR/SSIM: 360p@24fps: MPD, Segments 360p@6, 12, 24fps: MPD, Segments 720p@24fps: MPD, Segments 720p@6, 12, 24fps: MPD, Segments 1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments	1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments PSNR/SSIM: BL, EL1, EL2, EL3	1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments PSNR/SSIM: BL, EL1, EL2, EL3, EL3
<b>Short Sequences from <a href="http://media.xiph.org">media.xiph.org</a></b>					
Blue Sky YUV420p, 1024p, 24 fps	217 frames 5 segments	360p@24fps: MPD, Segments 360p@6, 12, 24fps: MPD, Segments 720p@24fps: MPD, Segments 720p@6, 12, 24fps: MPD, Segments 1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments	360p@24fps: MPD, Segments 360p@6, 12, 24fps: MPD, Segments 720p@24fps: MPD, Segments 720p@6, 12, 24fps: MPD, Segments 1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments	1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments	1080p@24fps: MPD, Segments 1080p@6, 12, 24fps: MPD, Segments
Rush Hour	500 frames	360p@24fps: MPD, Segments 360p@6, 12, 24fps: MPD, Segments 720p@24fps: MPD, Segments	360p@24fps: MPD, Segments 360p@6, 12, 24fps: MPD, Segments 720p@24fps: MPD, Segments	1080p@24fps: MPD, Segments	1080p@24fps: MPD, Segments

Figure 3.7. The Source of Videos

```

$> BitStreamExtractorStatic BBB-IV.264
JSVM 9.19.15 BitStream Extractor

Contained Layers:
=====

Layer  Resolution  Framerate  Bitrate  MinBitrate  DTQ
0      640x368    6.0000    162.50   162.50      (0,0,0)
1      640x368    12.0000   325.00   325.00      (0,1,0)
2      640x368    24.0000   650.00   650.00      (0,2,0)
3      1280x720   6.0000    550.00   550.00      (1,0,0)
4      1280x720   12.0000   1100.00  1100.00     (1,1,0)
5      1280x720   24.0000   2200.00  2200.00     (1,2,0)
6      1920x1088  6.0000    1250.00  1250.00     (2,0,0)
7      1920x1088  12.0000   2500.00  2500.00     (2,1,0)
8      1920x1088  24.0000   5000.00  5000.00     (2,2,0)
9      1920x1088  6.0000    2000.00  2000.00     (3,0,0)
10     1920x1088  12.0000   4000.00  4000.00     (3,1,0)
11     1920x1088  24.0000   8000.00  8000.00     (3,2,0)
12     1920x1088  6.0000    3750.00  3750.00     (4,0,0)
13     1920x1088  12.0000   7500.00  7500.00     (4,1,0)
14     1920x1088  24.0000  15000.00 15000.00     (4,2,0)

```

Figure 3.8. Example of DASH/SVC of BBB-I-360p Information

This open-source was used by M. Rotinsulu et al. [272] to measure the QoS in NDN by ndnSIM2.0 to simulate and evaluate [273]. The video model is a set of consecutive video chunks or segments,  $V = 1, 2, \dots, K$ , each of which contains  $L$  seconds of video is encoded with different bitrates. Thus, the total length of the video is  $K * L$  seconds. The video player can choose to download video segment  $k$  with bitrate  $R_k \in \mathbb{R}$ , where  $\mathbb{R}$  is the set of all available bitrate levels. The amount of data in segment  $k$  is then  $L * R_k$ . The higher bitrate is selected, the higher video quality is perceived by the user [177, 274]. The Figure 3.8 explains the information example of the DASH/SVC of BBB-I-360p.

### 3.4.2.2 Performance Metrics

The performance metrics are used to enhance and evaluate the performance of the proposed caching placement mechanism for NDN during VoD services. This attempts to improve overall QoS for multimedia services. Previously, an objective examination of the network was conducted by measuring many criteria to determine the quality of the network and quantify the QoS of the network. QoS refers to the network's ability to attain a more deterministic behaviour so that transportation of data is achieved with minimal loss of packets, delay, and maximum bandwidth. Note that QoS does not

consider the perception of the user. From experience, it is observed that a service provider cannot afford to wait to receive customer complaints in a situation where the customer experiences low-quality service. This is because, from the findings of an Accenture survey [84], about 90% of users will not register a complaint regarding the low QoS but would rather ditch the service provider and move on to another [275]. The selected metrics for measuring the performance of the new mechanisms are the average hit ratio, the average E2EDelay time, server load reduction, and total reduction in the network footprint.

Various factors are capable of affecting the perceived quality, and these include network reliability, the process of content preparation, and terminal performance. The content preparation process contains the caching placement mechanism, while the terminal performance is affected, especially on VoD services, which are considered a part of QoS. M. Alreshoodi et al. [276] focused on several interdependent parameters that determine the QoS of multimedia streaming services over networks. The researcher showed four performance indicators for the average hit ratio, the server load reduction, the average total reduction in the network footprint, and delay speed, which affects the user performance [277, 106, 104]. J. Auge et al. [278] analysed the performance metrics of the total reduction in the network footprint and proved that an increase in the number of requests in the network has an adverse effect on performance.

### **1. Average Hit Ratio**

The average hit ratio represents the proportion of content requests that are successfully fulfilled from the cache as opposed to being fetched from the original source, such as the server. "Hit" occurs when the requested content is found in the cache, while "miss" occurs when it must be retrieved from the source. In coordinated cache placement, content is strategically distributed across various cache nodes based on

factors like user demand patterns, network topology, and content popularity. The goal is to maximize the probability that requested content will be found in a nearby cache, thereby increasing the average hit ratio [120].

Moreover, a higher average hit ratio reduces the need for content to be retrieved from the origin server, which decreases overall network traffic, reduces latency, and minimises the load on the network infrastructure. In addition to quicker content retrieval, it reduces buffering times and improves the overall viewing experience for users. However, this high hit ratio leads to the network experiencing less load, particularly during peak times. This reduces congestion and helps maintain consistent performance levels even as the demand for VoD services increases. This leads to a more efficient use of network resources and improves QoS for users.

In NDN, video content is often partitioned into smaller, which are then individually cached. This allows for more granular control over caching and increases the probability that at least some of the requested videos are available in the cache. As VoD services scale and the number of users increases, maintaining a high average hit ratio ensures that the network can handle more simultaneous requests without degradation in QoS. Therefore, coordinated cache placement can dynamically adapt to changes in user behaviour, content popularity, and network conditions, continuously optimising the caching strategy to maintain a high average hit ratio [279].

This performance metric displays the potential of the caching mechanisms to initiate a moderation of the number of redundant content copies. The average cache hit ratio across all routers is computed from the following equation [72, 280]:

$$AverageHitRatio = \frac{\sum_{i=1}^n \frac{Hit_i}{Hit_i + Miss_i}}{n} \quad (3.2)$$

Where  $Hit_i$  is the number of requests that provided the data content by the router ( $r$ ) from its cache, and  $Miss_i$  is the number of requests which were forwarded upstream by  $r$  as a result of a cache miss.  $n$  represents the total number of routers. A higher average hit ratio indicates efficient caching, leading to reduced network traffic and latency [247].

The average hit ratio is a critical metric for assessing the effectiveness of coordinated cache placement in NDN during VoD services. It directly impacts network efficiency, user experience, and the service's scalability. By optimising the cache placement strategy and effectively partitioning video content, NDN can achieve a higher average hit ratio, leading to faster content delivery, reduced network load, and an overall better QoS for users.

## 2. Average E2EDelay

End-to-End Delay (E2EDelay) is a critical metric performance statistic that indicates a packet's overall latency while travelling across the network from its source to its destination. The total of the node delay at each node and the link delay at each link on the network are used to determine the end-to-end (E2E) delay path. This includes all delays due to data transmission, processing, queuing, and routing across the network. The strategy used to place content within caches across the network has a direct impact on E2EDelay. Coordinated cache placement involves strategically placing video content or its segments in locations that optimise retrieval times, which

minimises E2EDelay. Coordinated caching ensures that video segments are stored closer to where they are most likely to be requested. This reduces the distance data needs to travel, thereby lowering the average E2EDelay [120, 281].

In VoD services, a low average E2EDelay is crucial for a high-quality user experience, as it directly affects video load times, buffering, and the smoothness of playback. For VoD, where users expect instantaneous access to video content, the average E2EDelay is a critical metric. Coordinated caching in NDN can help ensure that videos are delivered quickly, improving the overall QoS. Partitioning videos in NDN leads to more efficient caching and faster retrieval. Coordinated cache placement ensures that the right content is available at the right locations, minimizing the time required to deliver each content during video playback.

According to the link delay, which only relates to the propagation delay on the link  $(i, j)$ . This link refers to the connection between node  $i$  and node  $j$  in the network. When considering a wireless link, the assumption is that each hop along the path has almost identical propagation delays [282]. A delay time is the disparity between a client time of Request  $R_i$  and the time of the server or source  $S_i$  to request number  $i$ . However, the average of E2EDelay is a summation of delay time divided by the total number of requests  $n$ , as follows [283, 284]:

$$\text{AverageE2EDelay} = \frac{\sum_{i=1}^n (R_i - S_i)}{n} \quad (3.3)$$

Average E2EDelay is an important performance metric in NDN for VoD services, particularly when using coordinated cache placement. It directly influences the QoS by

affecting how quickly and smoothly video content can be delivered to users. Through strategic partitioning of video content and intelligent placement within the network, coordinated caching can significantly reduce the average E2EDelay, leading to a better user experience with faster video load times and minimal buffering.

### **3. Server Load Reduction**

Server load reduction is a critical performance metric in evaluating coordinated cache placement, particularly in the context of NDN during VoD services. It refers to the decrease in the amount of data that needs to be served directly from the origin servers. This reduction is achieved when a higher proportion of content requests are fulfilled by caches distributed throughout the network rather than by the central server. Coordinated cache placement strategies aim to distribute video content across the network's caches strategically. By doing so, they reduce the frequency with which the origin server is accessed for content retrieval, thereby decreasing the server's load. This leads to improved overall network efficiency, as the distributed caches can handle a significant portion of content requests [279].

In NDN, video content is typically divided into smaller segments or chunks, allowing individual segments to be served from nearby caches rather than the central server. Therefore, this distributed caching approach significantly reduces the burden on the server, as most content is retrieved locally from the network caches. Thus, VoD services can handle more users and higher traffic volumes without degrading performance. This scalability is crucial for maintaining a high-quality user experience as the demand for VoD services grows [151, 285].

Reducing the load on the origin server is essential for enhancing the scalability, reliability, and efficiency of the network. It prevents bottlenecks, reduces the risk

of server overload, and ensures smoother operation, especially during peak demand times. This metric is on the origin servers, which are defined by the equation [72, 120] as follows:

$$\text{Server Load Reducion} = 1 - \text{NumCachedReq}_V \quad (3.4)$$

Where *NumCachedReq* denotes the number of video requests that were fulfilled using an in-network cache.

Server load reduction is a crucial metric for evaluating the effectiveness of coordinated cache placement in NDN during VoD services. By strategically placing video across the network's caches, the load on the origin server is significantly reduced, leading to improved scalability, reliability, and cost efficiency. This reduction in server load ensures that VoD services can deliver content quickly and efficiently to a large number of users, maintaining high levels of QoS even during peak usage times.

#### **4. Total Reduction in the Network Footprint**

The network footprint refers to the total amount of data traffic generated within the network as content is requested, transmitted, and delivered to end-users. Reducing the network footprint means that less data needs to travel through the network to deliver content to users. For VoD services, which transmit large volumes of video content, this efficiency is particularly crucial. Therefore, a smaller network footprint typically translates to lower congestion, faster delivery times, and more efficient use of network resources, all of which contribute to better QoS for end-users. Coordinated cache placement entails strategically distributing video content across multiple network caches. By placing content closer to the user, coordinated caching reduces the distance

that data needs to travel, which in turn reduces the network footprint [129].

Video partitioning in NDN enables more granular caching strategies, transmitting and caching only the necessary segments. NDN can minimize the amount of data that needs to be fetched from the original content source. This partitioning video further reduces the overall network footprint. A lower network footprint also means reduced costs for network operators, as there's less strain on network resources like bandwidth and fewer requirements for data transmission over long distances. In addition, the network can handle more simultaneous VoD requests without becoming congested, which makes the service more scalable. Therefore, faster content delivery and fewer network delays directly enhance the user experience, as videos load quicker and stream more smoothly.

The total number of byte-hops stored by in-network caching is defined by the following equation [72]:

$$TotalReductionNetworkFootprint = \frac{\sum_{r=1}^V (HopsNoCache_r - HopsNDN_r) * bytes_r}{\sum_{r=1}^V (HopsNoCache_r * bytes_r)} \quad (3.5)$$

Where  $HopsNoCache_r$  presents the total number taken by the request  $r$  to reach the closest origin server.  $HopsNDN_r$  presents the number of hops taken by the request  $r$  to arrive at the nearest location containing a copy of the video content, i.e., either an in-network cache or the origin server. The  $bytes_r$  refer to the size of the request  $r$ , which is presented in *byte* units. Furthermore,  $V$  is the total number of requests.

Total Reduction in the Network Footprint is a vital performance metric in the context of coordinated cache placement, particularly for VoD services in NDN. By strategically

placing video closer to end-users and reducing unnecessary data transmission across the network, coordinated caching significantly lowers the network footprint. This reduction leads to more efficient use of network resources, lowers operational costs, and enhances the overall QoS by reducing latency and improving the scalability and responsiveness of VoD platforms.

### **3.4.3 Confidence Level of Simulation Results**

Performance evaluation aims to determine the average ( $\mu$ ) of the performance measures previously covered, including average hit ratio, average E2EDelay, server load reduction, and overall network footprint reduction. In simulation experiments, random occurrences are produced using random numbers. This randomness causes random occurrences to also appear in the results. As a result, the output of numerous simulation runs determines the precision value of ( $\mu$ ). A degree of confidence in the result should be established because the simulation output is merely an estimate. The study employed ten runs with different random number generator seed settings for each situation. The point estimate for the averages of all collected measurements was calculated along with a 95% confidence range. Hasan and Jain [286], Bajaj et al. [287], and [288, 200, 289] provide additional details.

### **3.5 Summary**

The research mechanism to ensure that the goals for the study are indicated in Chapter One to ensure that the research objectives can be fulfilled, verified, and validated. This research focuses on developing a coordinated caching placement mechanism for NDN during VoD services, which attempts to improve QoS for VoD services. This will be accomplished by outlining the experimental design, implementation approach, and tool employed, as illustrated in Figure 3.1.

This chapter started with six research steps, which are: perform an in-depth study on the issues surrounding the coordination caching placement mechanism, perform an empirical analysis on the previous seven cache placement mechanisms, which are LCE, LCD, Cross, Prob, PProb, Centrality, and Rand mechanisms, to find the functionality requirements, design a novel coordinated caching placement mechanism by partitioned the requested video, CPVP, for use in NDN with VoD services, implement, validate, and verify the design of the novel mechanism, execute the experiments, and lastly carry out performance evaluation of the novel mechanism by comparing it with the LCE, LCD, Prob mechanisms.

This study used the simulation approach to evaluate the novel mechanism, which many other researchers have employed in this area. For evaluation metrics, this study chose to use the average hit ratio, average E2EDelay, server load reduction, and total reduction in the network footprint.

The experiments are performed using the ndnSIM 2.0 simulation approach. The data is VoD traffic obtained randomly from the SVC-DASH, as explained in Subsection 3.4.2.1.

This chapter has also explained how the results are verified and validated. All the algorithms have been inspected so that they meet the intended or the original requirements of the designed objectives.

The next chapter comprehensively discusses the empirical analysis of the seven placement caching mechanisms. By thoroughly understanding their strengths and weaknesses in functional requirements, the novel coordinated caching placement (CPVP) can be effectively designed.

# CHAPTER FOUR

## THE EMPIRICAL EVALUATION OF CACHING PLACEMENT MECHANISMS

This chapter aims to conduct an empirical evaluation of caching placement mechanisms to identify the functional requirements for Named Data Networking (NDN) coordinated caching placement performance mechanisms for VoD workloads. This chapter presents the results of implementing placement mechanisms LCE, LCD, Prob, PProb, Rand, Centrality, and Cross during VoD. To achieve the first goal of **Chapter One**, the chapter also evaluates the performance metrics outcomes that are related to the aforementioned procedures.

### 4.1 Introduction

The literature on Information-Centric Networking (ICN) has extensively explored and investigated architecture improvement measures [106, 168, 31]. This chapter will explore and analyse metrics for the NDN designs, such as average cache hit ratio, average End-to-End Delay (E2EDelay), origin server load reduction, and network footprint reduction. To empirically evaluate the placement mechanisms to identify the functional requirements needed for NDN-coordinated caching placement performance mechanisms for VoD workloads, metrics could be optimised using a variety of caching-related factors, as well as content placement. Based on these functional requirements, a new Coordinated Cache Placement by Video Partitioning (CPVP) mechanism will be designed as a second objective in NDN for VoD services. CPVP is achieved by combining three new schemes, which are VAS to deal with content retrieval efficiency, VCDS to optimise caching decisions, and VPS to enhance dynamic partitioning.

This work examines the effects of various NDN caching placement mechanisms using key performance metrics. The seven cache placement mechanisms are evaluated in

four cache sizes of *1GB*, *10GB*, *100GB*, and *1TB* with different scenarios to achieve the first objective, which will lead to designing a new cache placement mechanism in the next chapter.

## 4.2 Configuration of the Mechanisms

The researcher considers the typical arrangement depicted in Figure 4.1 because the configuration of placement mechanisms LCD, LCE, and Prob shares some characteristics. Whereas LCE is referred to as an example of non-coordinated caching placement, LCD and Prob are referred to as examples of implicit coordinated caching placement. It is commonly known that most multi-level caches currently operate typically using the LCE caching placement mechanism. For this study, the researcher configures the LCE to provide a copy of the requested item when it is present in all intermediate caches (levels  $l - 1, \dots$ , and  $l$ ) between the location of the hit and the requesting client. This is configured when a hit happens at the origin server or a level  $l$  cache.

In this investigation, the LCD caching placement method was configured only to store a fresh duplicate of the requested object in the  $(l - 1)$  – level cache. This was set up so the cache could find the requesting client by positioning it below the hit. Because it makes several requests to add things to a cache, the LCD structure is comparable to the LCE. Additionally, it was programmed to find every request by making a fresh replica of the item one hop lower.

The Rand caching placement method properties are set to choose one of the path's nodes randomly. It was set up to store a duplicate of the content on a randomly selected node while its reference packet is processed over the network to establish the potential field for node caching. In the simulation, the researcher looked at sending random requests within a specified range to the relevant node to distribute the content file and

the inquiry. This was required to enable the random forwarding of queries within some nodes until the potential field was found.

When using the Prob caching placement technique, the researcher configures the cache placement by managing the connections between the referred query, the client's intermediate cache, and the path from the hit location. The researcher suggests putting a reference object there to help trace the client's request back to its source. On the other hand, a local copy from the intermediate cache is preserved with probability  $p$ ; as a result, the replacement mechanism is activated. This suggests that it is unlikely that a copy of the current task was put in the interim cache  $1 - p$  [156].

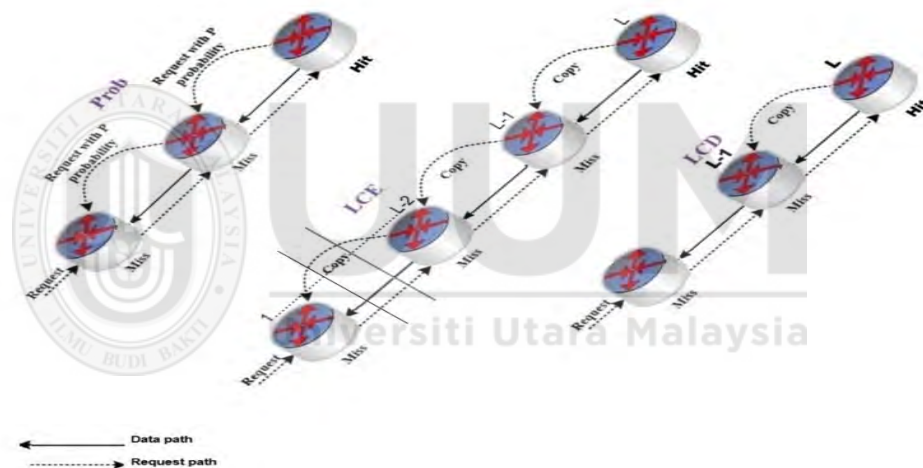


Figure 4.1. The Operation of the LCD, Prob, and LCE Placement Mechanisms [156]

The PProb caching mechanism is a probabilistic approach for distributed content caching along a patch of caches. In this work, the researcher sets up PProb to calculate the caching capability using the probabilistically proposed node path location and cache contents [156], as in Figure 4.2. This frees up space in the cache for other flows that use the same path and fairly multiplexes the contents of different flows among caches on the same path.

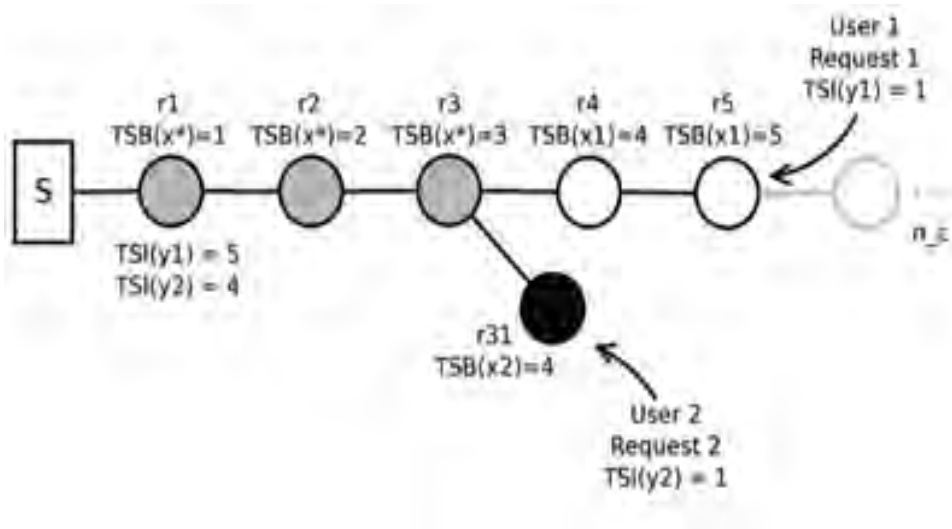


Figure 4.2. The PProb Caching Mechanism Scenario

This work employed the centrality-based caching mechanism to increase caching gain and lessen placement performance uncertainty. By considering the connection between pairs of nodes in the network, the researcher examined the configuration proposed by Chai, He, Psaras, and Pavlou [159] to specify the frequency of preset nodes on the content delivery path. In this study, the researcher explained cache misses inside a network by assuming the content node would fail with a sizable number of content delivery channels. The researcher guaranteed that the cache replacement rate was decreased in addition to caching content by only caching nodes with higher priority.

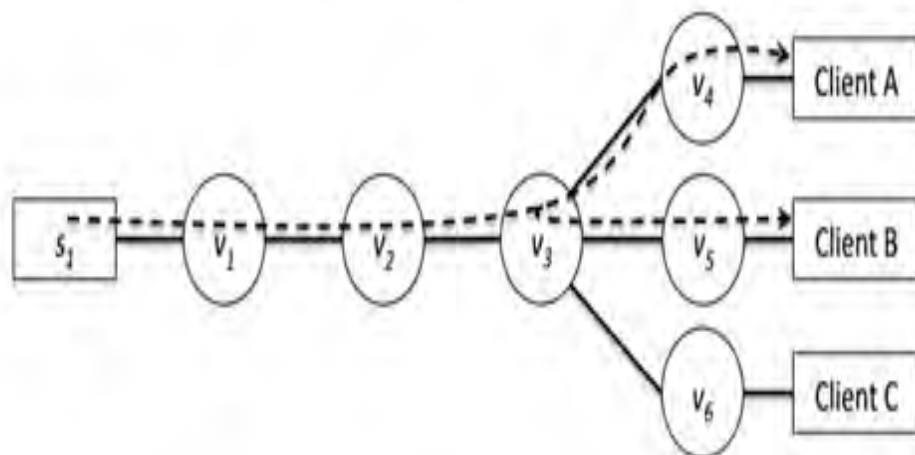


Figure 4.3. Centrality-based Caching Mechanism Node Positioning

By considering the connections between the network topology and the material's popularity, this study set the cross-caching placement mechanism in a cross-layer form to cache contents in a few chosen locations. The mechanism makes use of information that is accessible at both the network and application levels.

### 4.3 Simulation Scenario

In this study, the implementation will be simulated seven specific content placement mechanisms: LCD, LCE, Prob, PProb, Centrality, Rand, and Cross. This simulation will use various cache sizes (1GB, 10GB, 100GB, and 1TB). This diversity of cache size is commendable to show the relationship between increasing the cache size and decreasing the delay and overload of the network. Whereas VoD applications utilise the cache size. Furthermore, the scenarios fail to address critical aspects such as dynamic user demands or network congestion, which are common in VoD environments. Incorporating these factors would significantly enhance the study's applicability and robustness. Whereas the requests' probability distribution of all content files follows Zipf distribution with  $\alpha = 0.65$  and  $\alpha = 2.0$  using a default replacement mechanism, Last Recently Used (LRU). Each result was obtained by averaging over ten simulation runs.

However, these caching placement mechanisms will be simulated using ndnSIM 2.0 and the Abilene topology as in Figure 4.4. The foundation of the Abilene topology is a link between directly connected nodes arranged in a bidirectional order, each serving various functions and whose link utilisations typically varies significantly.

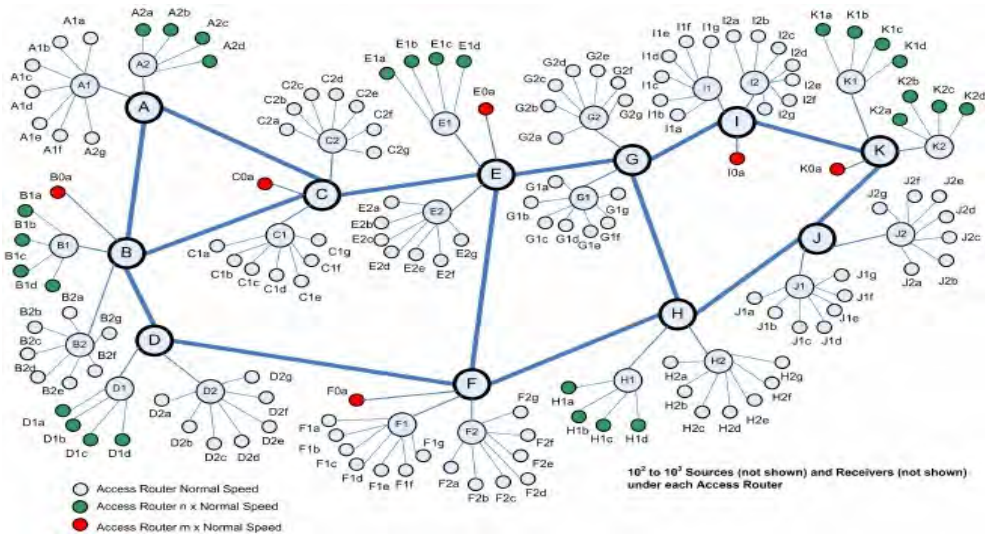


Figure 4.4. Abilene Topology

This study evaluates the seven caching placement mechanisms stated in the preceding section using VoD. This evaluation is measured by using the quality metrics of the played video or its subjective perception by the end user to quantify QoE (Quality of Experience). The most crucial QoE measures affect user engagement [173]. These parameters are the buffering ratio, average bitrate, and startup latency.

#### 4.4 Simulation Results and Evaluation of Seven Placement Mechanisms

The QoE provided across various environmental circumstances can be improved by several trends connected to the performance of networking applications [174, 290]. Because it offers a thorough grasp of the data reduction state, the researcher in this study considered specific network measurements when building the PProb mechanism's network metrics. Determining the root cause of this delay can compromise the entire network's performance. Therefore, it is advisable to address the summary value produced from the delay measurements on a single domain's links from the viewpoints of an average cache hit ratio, a total reduction in network footprint, and a decrease in server load.

#### 4.4.1 Average Hit Ratio

The following diagrams show the average cache hit ratio for the seven cache placement mechanisms via different sizes  $1GB$ ,  $10GB$ ,  $100GB$ , and  $1TB$  using  $\alpha = 0.65$ , then  $\alpha = 2.0$  as explained previously. The equation 3.2 was used in this investigation to get the average cache hit ratio across all routers:

##### 4.4.1.1 Average Hit Ratio Using 1 GB and $\alpha = 0.65$

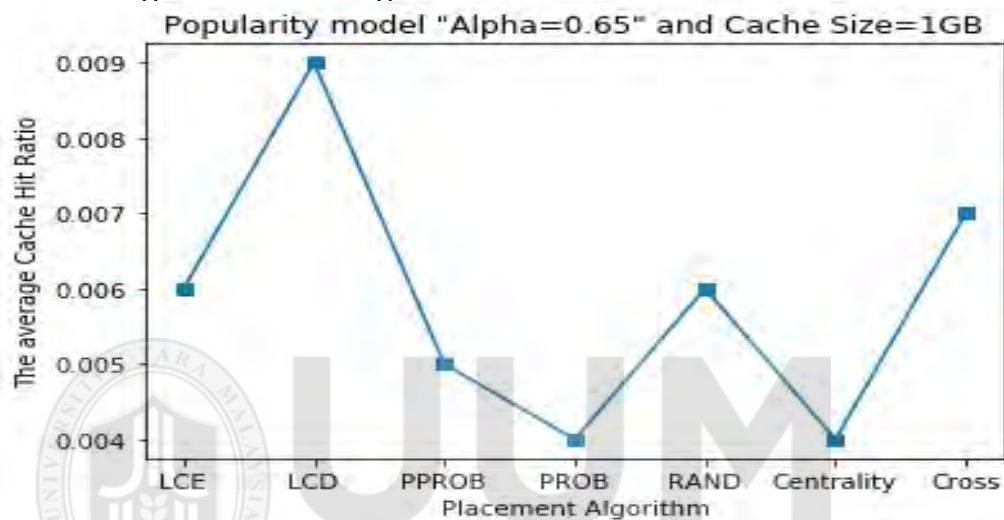


Figure 4.5. Average Cache Hit Ratio for Seven Placement Mechanisms Using 1 GB and  $\alpha = 0.65$

Figure 4.5 shows the average cache hit ratio for all mechanisms whenever the cache size is 1GB. Prob and Centrality, however, share a 0.004 average cache hit ratio. The other mechanisms range in size from 0.005 to 0.009 simultaneously. Finally, the outcome revealed that among the 1GB methods, LCD had reserved the maximum ratio of 0.009.

Although LCD stores the content on each node on the level below where the hit occurs, LCD outperforms LCE since LCE stores a duplicate of the content on each node along the path in its cache. Additionally, semi-structures were employed by LCD and LCE since it takes numerous queries to add objects to a leaf cache [16]. Further, they

decided to track each request by making a fresh duplicate of the object one client's hop away due to increase the video bitrate. Higher bitrates typically mean better video quality, as more data can be transferred per second, improving the viewing experience. Additionally, it was emphasised by Nikolaos L. et al. [163] that the LCD is superior to LCE since it can provide exclusive caching. In other words, the LCD prevents the repetitive repetition of the same few things by allowing each cache in a chain of stores to retain a potentially diverse collection of objects.

#### 4.4.1.2 Average Hit Ratio Using 10 GB and $\alpha = 0.65$

The cache size is raised to 10GB in Figure 4.6. Here, we can immediately see that the LCD mechanism's average cache hit ratio performs best [264]. Its average cache hit ratio is 0.023, higher than Rand's (0.017). While Centrality produced the lowest ratio of 0.009, other methods like LCE, Prob, PProb, and Cross achieved interchangeable ratios. The average cache hit ratio is once more lowest when the centrality approach is used.

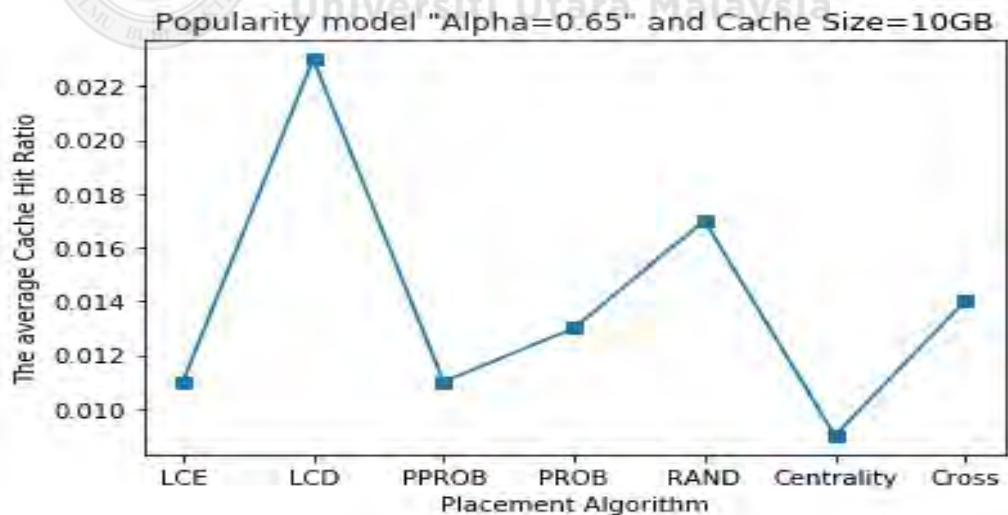


Figure 4.6. The Average Cache Hit Ratio for Seven Placement Mechanisms Using 10 GB and  $\alpha = 0.65$

#### 4.4.1.3 Average Hit Ratio Using 100 GB and $\alpha = 0.65$

The cache size is raised to 100GB in Figure 4.7. It is clear that the Rand method, with an average cache hit ratio of 0.048, performs best, with Prob and LCD coming in second and third, respectively. In contrast to Cross, LCE, and PProb, other mechanisms, such as Centrality, produced a relatively low ratio of 0.024.

However, the RAND method outperforms other mechanisms because it is randomly selects items for eviction when space is needed. At first glance, RAND might seem suboptimal compared to more intelligent, decision-based policies. However, there are scenarios where RAND can be effective. In addition, if the access pattern to the cache is approximately uniform, meaning all items have an equal probability of being accessed, a random eviction policy might perform similarly to other, more complex policies. In environments where content popularity changes rapidly and is hard to predict, making 'intelligent' decisions based on recent history can sometimes be misleading. In such cases, a random choice avoids the pitfalls of making decisions based on possibly outdated or irrelevant data. RAND is simple and doesn't require maintaining additional metadata or complex decision-making algorithms. This can result in faster operations and reduced overhead, which is especially in high-speed or resource-constrained environments. RAND doesn't favour any particular set of data, ensuring that all data pieces have an equal chance of remaining in the cache. This can be seen as a form of fairness, especially in multi-tenant environments. Intelligent caching policies can sometimes be tricked by crafty access patterns, leading to cache pollution where the cache is filled with seldom-used items. Random caching is immune to such patterns and ensures a level of unpredictability. In research or experimental settings, RAND can serve as a baseline to measure the effectiveness of other caching policies. However, it's essential to note that the success of RAND versus other caching policies largely depends on the specific application and workload. In many real-world

scenarios, especially with discernible access patterns, policies like Least Recently Used (LRU) tend to outperform RAND.

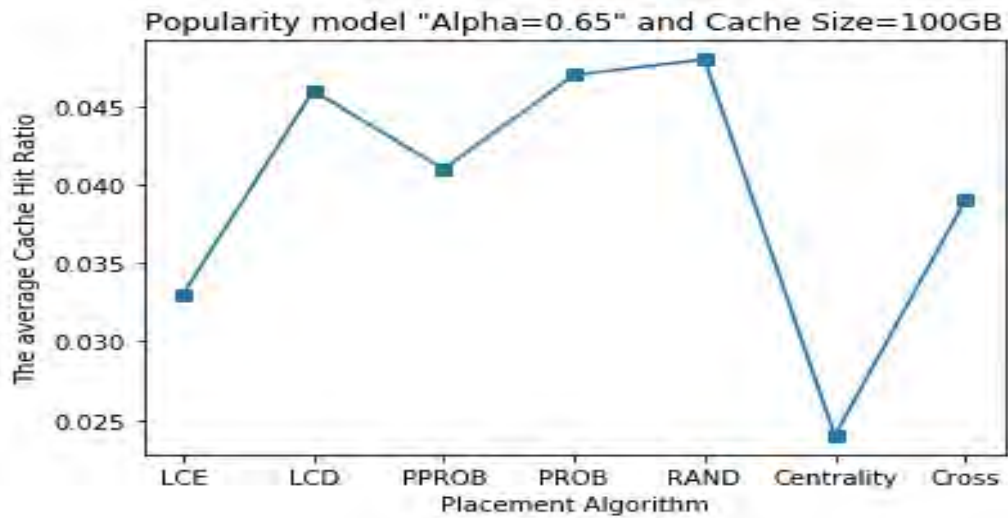


Figure 4.7. Average Cache Hit Ratio for Seven Placement Mechanisms Using 100 GB and  $\alpha = 0.65$

#### 4.4.1.4 Average Hit Ratio Using 1 TB and $\alpha = 0.65$

When the cache is 1TB in size, the average cache hit ratio for all seven caching placement mechanisms is shown in Figure 4.8. Prob then, Rand has the greatest average cache hit ratio compared to the other six mechanisms, with values of 0.095 and 0.091, sequentially. The average cache hit ratio of the PProp, Cross, and LCE mechanisms is marginally lower than that of the Prob mechanism. With an average cache hit ratio of 0.043, the centrality mechanism once more produces the poorest results.

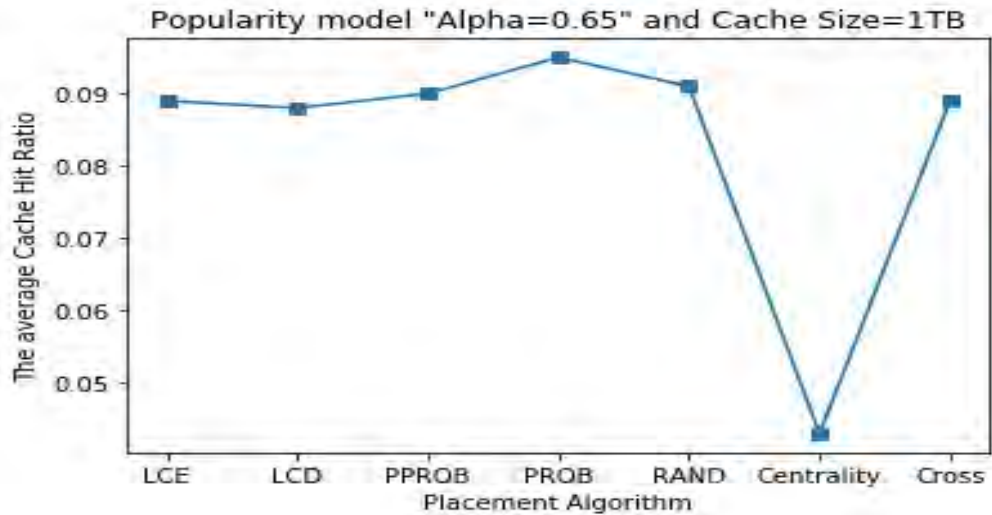


Figure 4.8. Average Cache Hit Ratio for Seven Placement Mechanisms Using 1 TB and  $\alpha = 0.65$

#### 4.4.1.5 Average Hit Ratio Using 1 GB to 1 TB and $\alpha = 0.65$

From Figure 4.5 to Figure 4.8, it can be seen that the average cache hit ratio rises as the cache size increases in all cases for all seven caching placement mechanisms. The LCD mechanism gives the best results for the average cache hit ratio from 1GB to 100GB [16, 264], while Prob is the best in 1TB. The centrality mechanism was the worst, as shown in Table 4.1. Moreover, the PProb mechanism keeps more than the middle level. Also, the performance of PProb will be better when the cache size increases. However, the figure 4.9 explains briefly Table 4.1.

**Table 4.1**

The Average Cache Hit Ratio for Seven Placement Mechanisms Using 1 GB to 1 TB and  $\alpha = 0.65$

Cache Size	LCE	LCD	PProb	Prob	Rand	Centrality	Cross
1GB	0.006	0.09	0.005	0.004	0.006	0.004	0.007
10GB	0.011	0.023	0.011	0.013	0.017	0.009	0.013
100GB	0.033	0.046	0.041	0.047	0.048	0.024	0.038
1TB	0.89	0.088	0.09	0.095	0.091	0.043	0.085
Average	0.03475	0.0415	0.03675	0.03975	0.0405	0.02	0.03525

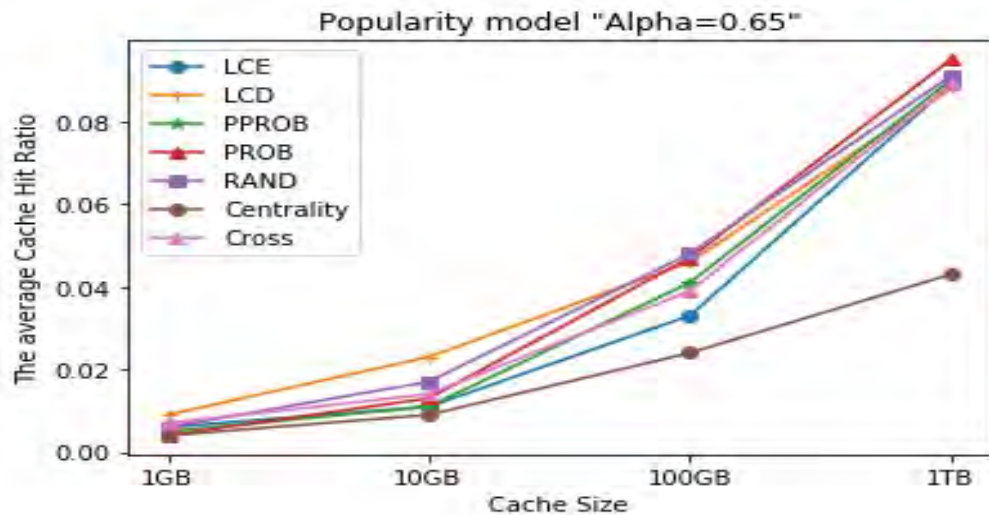


Figure 4.9. The Average Cache Hit Ratio in Four Cache Size for Seven Placement Mechanisms with  $\alpha = 0.65$

The earlier studies [290, 267] concluded that the cache mechanisms roughly performed an optimal distribution of cache content by reducing cache content redundancy (nevertheless, LCD has reduced redundancy compared to LCE) [264], which increases the variety of content mechanisms. As a result, the hit ratio was effectively increased. The research [72, 267, 291] also discovered that the LCD had a greater average cache hit ratio than LCE.

The figure displays the average cache hit ratio for the PProb mechanism with various cache sizes, ranging from 1GB to 1TB. The intermediate cache hit ratio rises in direct proportion to cache size [48]. Because the PProb mechanism can make the network store multiple copies of the same content request, therefore, with the increase of the node cache capacity, the average cache hit ratio is significantly increased.

Overall conclusion: This study necessitates a proposed placement mechanism that can satisfy each mechanism's positive effects while avoiding the negative. Therefore, the proposed mechanism needs to cache video close to the consumer, duplicate the caching without wasting storage through redundancy, and partition the video across multiple

routers to increase the average hit ratio.

#### 4.4.1.6 Average Hit Ratio Using 1 GB to 1 TB and $\alpha = 2.0$

The average cache hit ratio for all mechanisms in four cache sizes, as in Table 4.3 are explained in Figure 4.10. Therefore, it can be concluded that Centrality and LCE mechanisms have the worst average hit ratio values in the 1GB to 1TB cache sizes. While the Prob and Rand mechanisms then, PProb obtains the best overall cache sizes if focusing on average results. The results show that the average hit ratio performance metric increases when the cache size increases. These placement mechanisms, combining topology awareness and content popularity, offer the highest scalability for VoD workloads [72]. At the same time, when the Zipf distribution is increased from  $\alpha = 0.65$  to  $\alpha = 2.0$ , the average cache hit ratio performance metric is also increased.

**Table 4.2**

The Average Cache Hit Ratio for Seven Placement Mechanisms Using 1 GB to 1 TB and  $\alpha = 2.0$

Cache Size	LCE	LCD	PProb	Prob	Rand	Centrality	Cross
<b>1GB</b>	0.01	0.017	0.011	0.009	0.011	0.009	0.015
<b>10GB</b>	0.021	0.041	0.023	0.028	0.033	0.019	0.027
<b>100GB</b>	0.062	0.091	0.084	0.093	0.093	0.047	0.081
<b>1TB</b>	0.165	0.177	0.178	0.182	0.177	0.087	0.163
<b>Average</b>	0.0645	0.326	0.074	0.078	0.0785	0.0405	0.0715

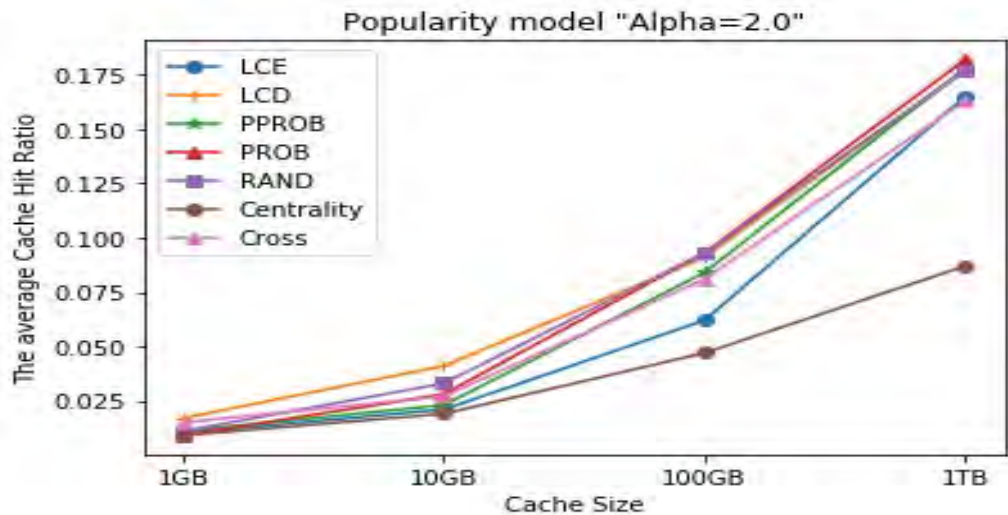


Figure 4.10. The Average Cache Hit Ratio for Seven Placement Mechanisms Using 1 GB to 1 TB and  $\alpha = 2.0$

#### 4.4.2 Average E2EDelay

There are delays between the client's time request and the server's time response. The average E2EDelay is consequently determined as in Equation 3.3 [292]. The average E2EDelay performance of seven mechanisms in various cache sizes—1GB to 1TB—is analysed and contrasted. Additionally, the comparison used the default configuration, which viewed the sent packet as the primary data transfer source.

##### 4.4.2.1 Average E2EDelay Using 1 GB and $\alpha = 0.65$

Figure 4.11 shows the delay performance for all mechanisms. The Centrality, CROSS, and RAND methods produce high delay performances of more than 4.82 sec. However, LCE had the best delay performance, clocking in at 4.782 sec, ahead of Prob (4.785 sec) and PProb (4.792 sec). This performance of LCE is because the queue editor that comprises LCE sends the processed packets in a fixed order [72, 170]. This led the researcher to conclude that LCE, followed by Prob, is the best in terms of delay performance when caching using a 1GB size cache.

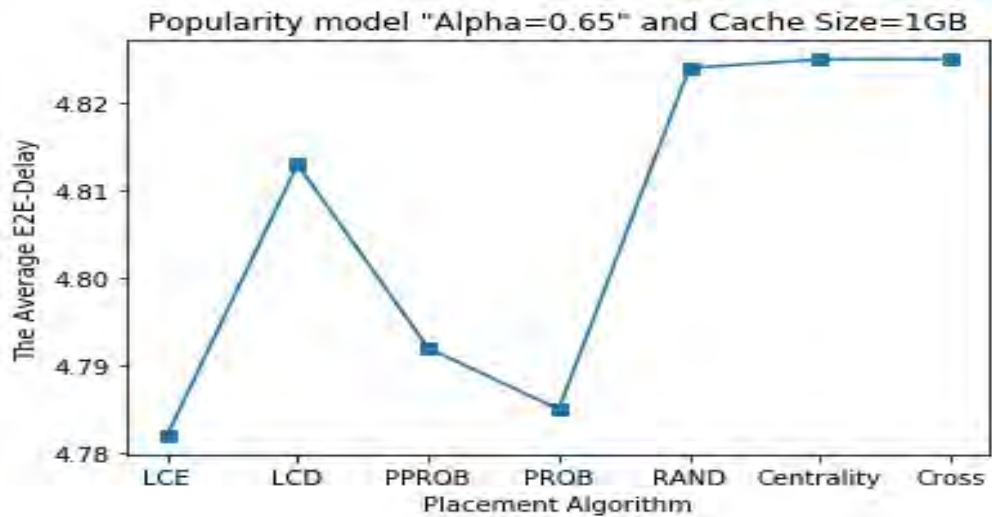


Figure 4.11. Average E2EDelay for Seven Placement Mechanisms Using 1 GB and  $\alpha = 0.65$

#### 4.4.2.2 Average E2EDelay Using 10 GB and $\alpha = 0.65$

The delay performance for all mechanisms is shown in Figure 4.12. One can see from the outcome that the same three mechanisms—CENTER, RAND, and CROSS—have once more produced high delay performances longer than 4.71 sec. However, PProb, LCE, and Prob were determined to have the lowest delay performances, with 4.669, 4.677, and 4.684 sec, respectively. Thus, PProb achieves the best results of the average E2EDelay by employing excellent queue management in larger cache sizes, where packets are processed based on priority and number of nodes [293, 156]. This led the researcher to conclude that PProb, followed by LCE, is enough to delay performance in a cache with a 10GB size because PProb has more diversity and less redundancy than LCE.

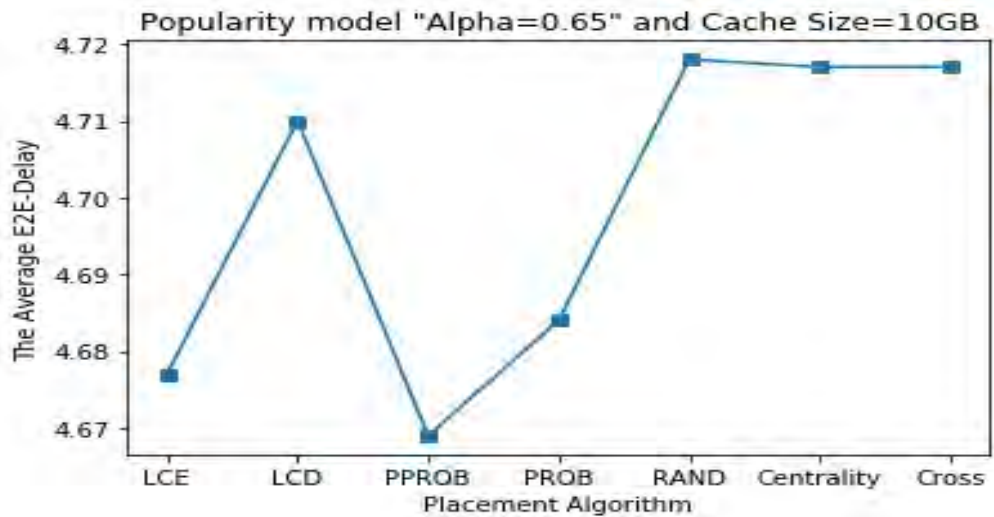


Figure 4.12. Average E2EDelay for Seven Placement Mechanisms Using 10 GB and  $\alpha = 0.65$

#### 4.4.2.3 Average E2EDelay Using 100 GB and $\alpha = 0.65$

Figure 4.13 displays the delay performance for all mechanisms. The performance results showed that the CROSS and RAND methods had high delay performance (delay =4.708sec and 4.683sec, respectively). However, PProb (4.599Sec) and Prob (4.564Sec) were discovered to have the lowest delay performances. Because Prob divides the packet into levels that the network can recall and process more quickly, it is useful for processing larger cache sizes [16, 156]. The researcher concluded that Prob and PProb are still regarded as adequate mechanisms for slowing down performance in 100GB caches as a result.

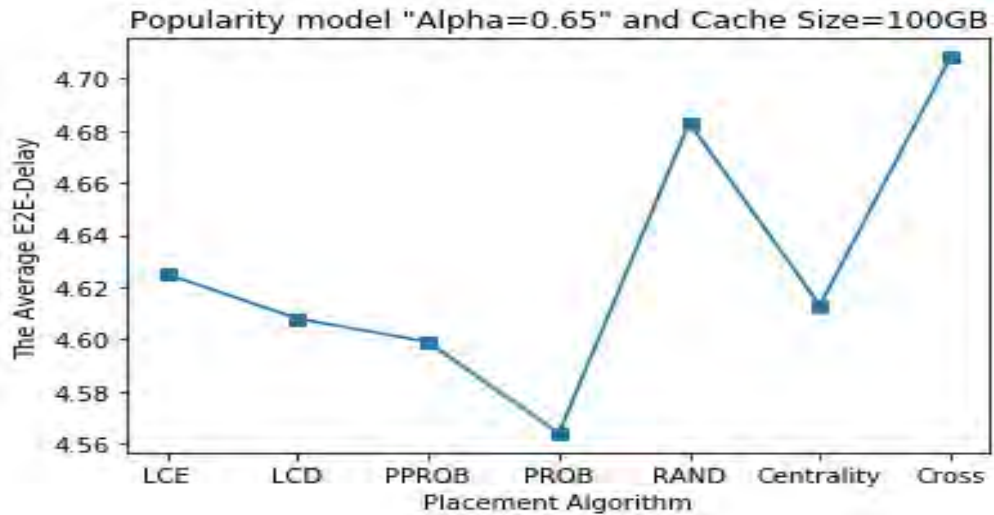


Figure 4.13. Average E2EDelay for Seven Placement Mechanisms Using 100 GB and  $\alpha = 0.65$

#### 4.4.2.4 Average E2EDelay Using 1 TB and $\alpha = 0.65$

The delay performance for all mechanisms is displayed in Figure 4.14 at 1TB of cache size. The performance results showed that CROSS and RAND, which use identical mechanisms, have great delay performance (delay = 4.638 sec and 4.613 sec, respectively). However, it was discovered that PProb (4.513Sec) and Prob (4.511Sec) had the lowest delay performances overall. This may be related to the previous argument, where Prob distributes the packet into levels that the network can recall and process more quickly [16, 156]. This led to the conclusion that Prob and PProb are still considered adequate mechanisms for slowing down performance in 1TB caches.

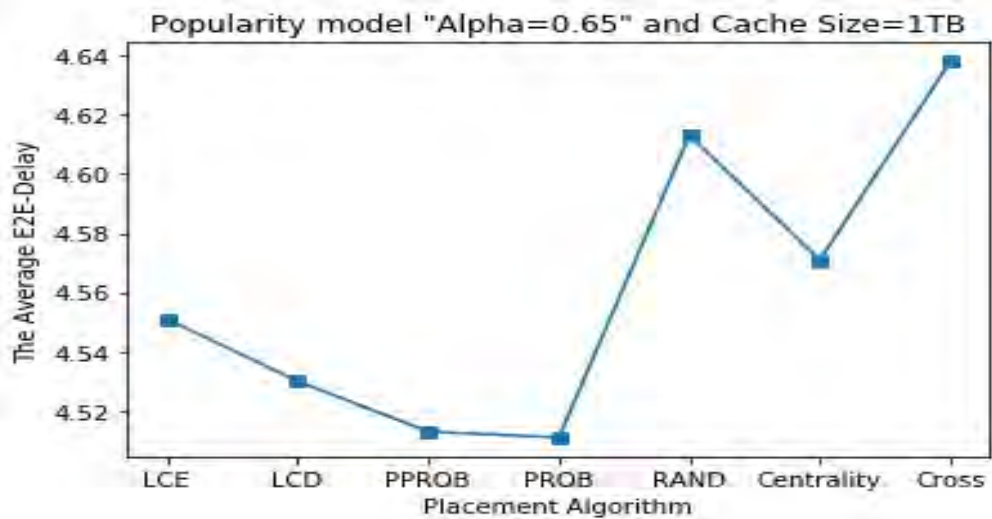


Figure 4.14. Average E2EDelay for Seven Placement Mechanisms Using 1 TB and  $\alpha = 0.65$

#### 4.4.2.5 Average E2EDelay Using 1 GB to 1 TB and $\alpha = 0.65$

Additionally, the delay for all mechanisms in four cache sizes is explained in Figure 4.15, which is also shown in Table 4.3. Therefore, in 1GB, LCE provides the best delay performance. At the same time, the CROSS and RAND mechanisms have approximately the worst delay values in the 1GB cache capacity. The PProb mechanism obtains the optimal method in 10GB concurrently. The least delay time in 100GB and 1TB is between Prob and PProb. But the causes of these delays are described in the previous section. The results demonstrate that as cache capacity is increased, the E2EDelay performance parameter improves.

**Table 4.3**

Average E2EDelay for Seven Placement Mechanisms Using 1 GB to 1 TB and  $\alpha = 0.65$

Cache Size	LCE	LCD	PProb	Prob	Rand	Centrality	Cross
<b>1GB</b>	4.782	4.813	4.792	4.785	4.824	4.825	4.825
<b>10GB</b>	4.677	4.71	4.669	4.684	4.718	4.717	4.717
<b>100GB</b>	4.625	4.608	4.599	4.564	4.683	4.613	4.708
<b>1TB</b>	4.551	4.53	4.513	4.511	4.613	4.571	4.638
<b>Average</b>	4.659	4.665	4.643	4.636	4.709	4.682	4.722

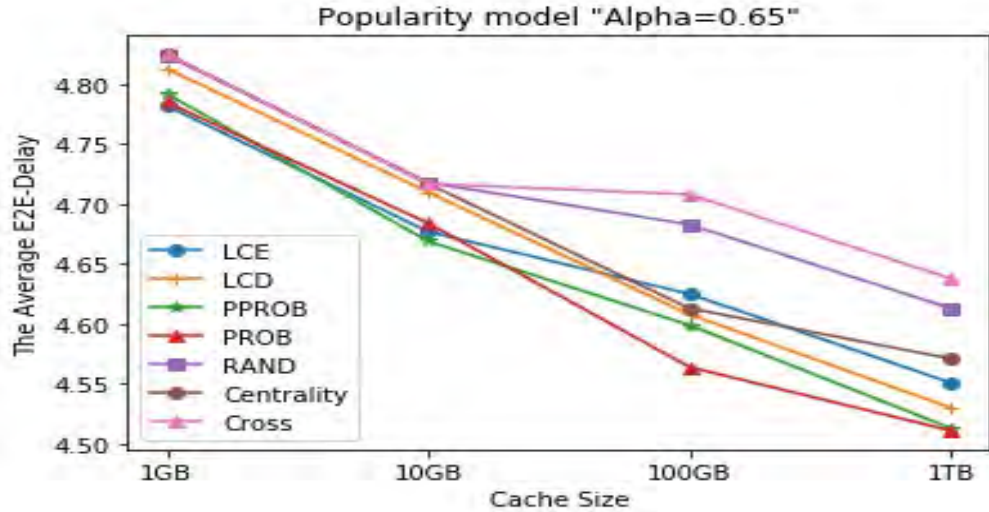


Figure 4.15. Average E2EDelay for Seven Placement Mechanisms Using 1 GB to 1 TB and  $\alpha = 0.65$

#### 4.4.2.6 Average E2EDelay Using 1 GB to 1 TB and $\alpha = 2.0$

Figure 4.15 elaborates the delay for all mechanisms in the four cache sizes as shown in Table 4.3. Therefore, it can be said that in the range of 1GB to 1TB cache sizes, the CROSS and RAND mechanisms have the worst latency values. When compared to other mechanisms, PProb achieves the best total cache sizes. The least delay time in 100GB and 1TB is between Prob and PProb. The results show that the E2EDelay performance metric is decreases when increasing the cache size, according to Che's approximation and content-centric networking principles [155], cache hit ratio improves with size, which inversely impacts delay [72]. At the same time, when the Zipf distribution is increased from  $\alpha = 0.65$  to  $\alpha = 2.0$ , the E2EDelay performance metric is decreased.

**Table 4.4**

Average E2EDelay for Seven Placement Mechanisms Using 1 GB to 1 TB and  $\alpha = 2.0$

Cache Size	LCE	LCD	PProb	Prob	Rand	Centrality	Cross
<b>1GB</b>	3.385	3.407	3.385	3.391	3.402	3.422	3.422
<b>10GB</b>	3.339	3.354	3.333	3.344	3.355	3.401	3.348
<b>100GB</b>	3.312	3.303	3.281	3.292	3.342	3.326	3.301
<b>1TB</b>	3.272	3.264	3.251	3.235	3.306	3.285	3.264
<b>Average</b>	3.327	3.332	3.313	3.316	3.351	3.359	3.334

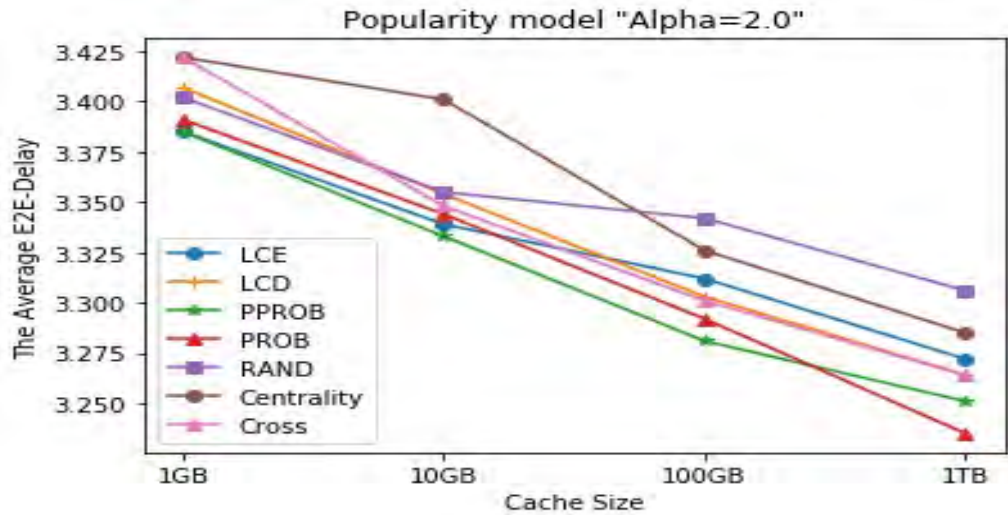


Figure 4.16. Average E2EDelay for Seven Placement Mechanisms Using 1 GB to 1 TB and  $\alpha = 2.0$

#### 4.4.3 Server Load Reduction

A critical performance metric is a decrease in load on the origin servers, especially when using an in-network cache like NDN. The server load reduction metric can be represented by the Equation 3.4. This subsection discusses the implementation results of the server load reduction for each of the seven cache placement mechanisms at various cache sizes.

##### 4.4.3.1 Server Load Reduction Using 1 GB and $\alpha = 0.65$

Figure 4.17 illustrates the performance of the reduction in server load for each of the seven mechanisms when the cache size is 1GB. The server load is significantly reduced when the LCD mechanism is utilized, whereas the Centrality mechanism results in a minor decrease. Consequently, the LCD mechanism yields optimal results by significantly easing the server load, thereby enhancing overall performance. Psaras et al. [16] confirm that LCD is one of the most effective strategies under constrained cache regimes, outperforming LCE and random approaches. The lower server load results in LCD performance due to a higher average hit ratio. However, the server load is minimize, when the request is hit in any routers before arriving at the server.

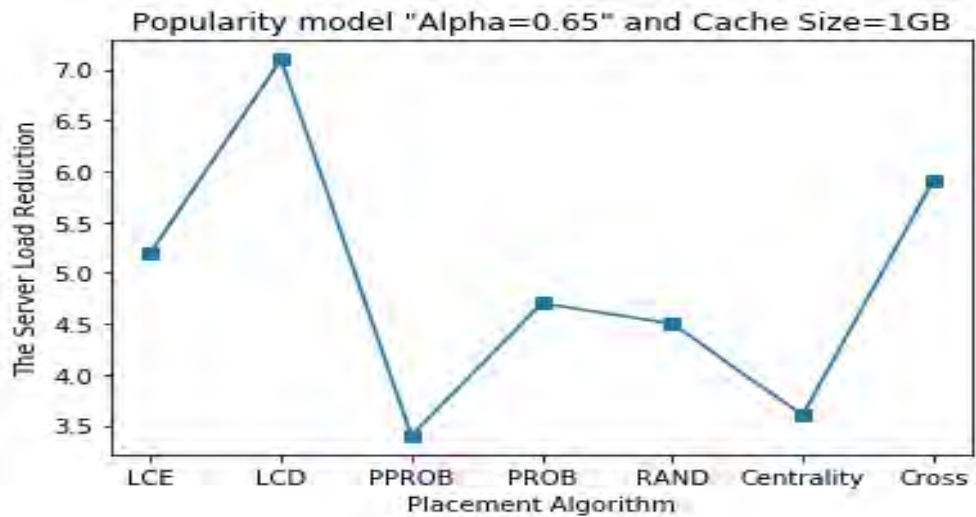


Figure 4.17. Server Load Reduction for Seven Placement Mechanisms Using 1 GB and  $\alpha = 0.65$

#### 4.4.3.2 Server Load Reduction Using 10 GB and $\alpha = 0.65$

When the cache size is 10GB, the results of reducing the server load for each of the seven mechanisms are shown in Figure 4.18. Here, the reduction in server load for PProp and Prob is the same (10.6%), considered the greatest. Other mechanisms, such as Cross, LCE, and LCD, delivered partial outcomes. In this instance, the centrality mechanism's value for the server load reduction is 7.9%, continuing its trend of producing the poorest outcomes.

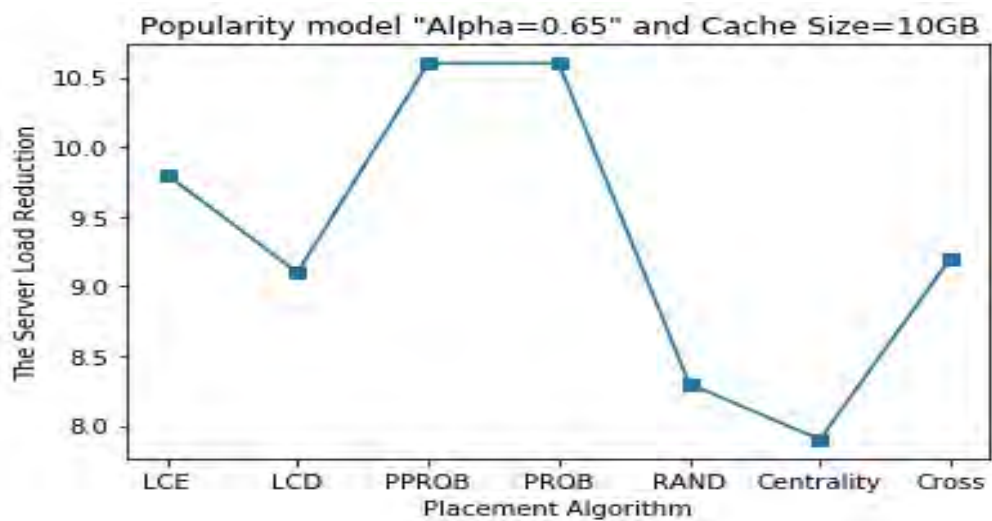


Figure 4.18. Server Load Reduction for Seven Placement Mechanisms Using 10 GB and  $\alpha = 0.65$

#### 4.4.3.3 Server Load Reduction Using 100 GB and $\alpha = 0.65$

Figure 4.19 presents the server load reduction results for seven placement mechanisms using 100GB and  $\alpha=0.65$ . The LCD method reduces server load by the maximum percentage, around 29.7%, while the Centrality mechanism performs at the lowest level, 16.2%. However, the centrality mechanism performs worse due to the presence of a bottleneck or loss link, as well as its higher distance from the consumer compared to the LCD method. Moreover, when repeating the same request, LCD places the content closest to the consumer through a single router.

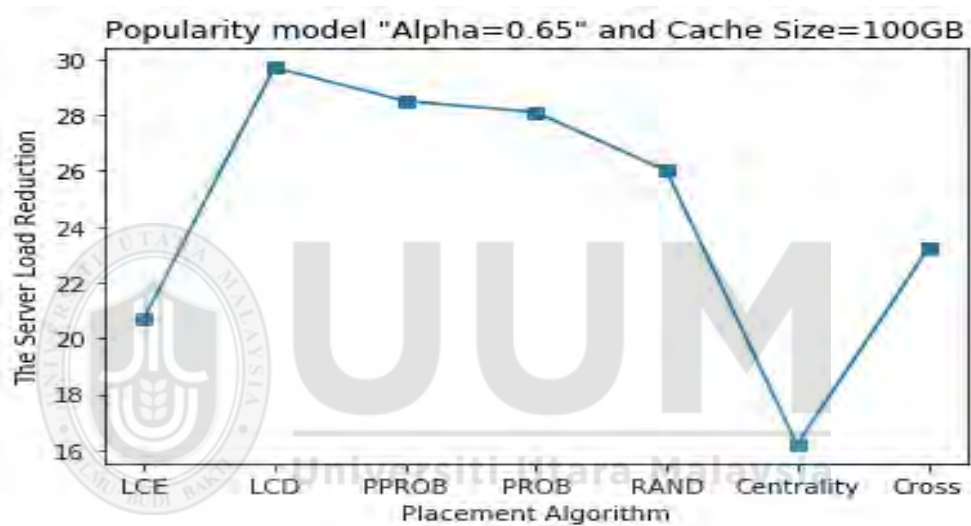


Figure 4.19. Server Load Reduction for Seven Placement Mechanisms Using 100 GB and  $\alpha = 0.65$

#### 4.4.3.4 Server Load Reduction Using 1 TB and $\alpha = 0.65$

The findings for seven mechanisms when the cache size is 1TB are shown in the Figure 4.20. The findings indicate that, at roughly 52.4%, the PProb mechanism has the largest impact on reducing server load, followed by the Prob mechanism (51.3%) because With more space, PProb avoids evicting frequently accessed items and can sustain high content availability locally, directly reducing requests to the origin server [294]. The lowest number, less than 25.8%, belongs to the centrality mechanism.

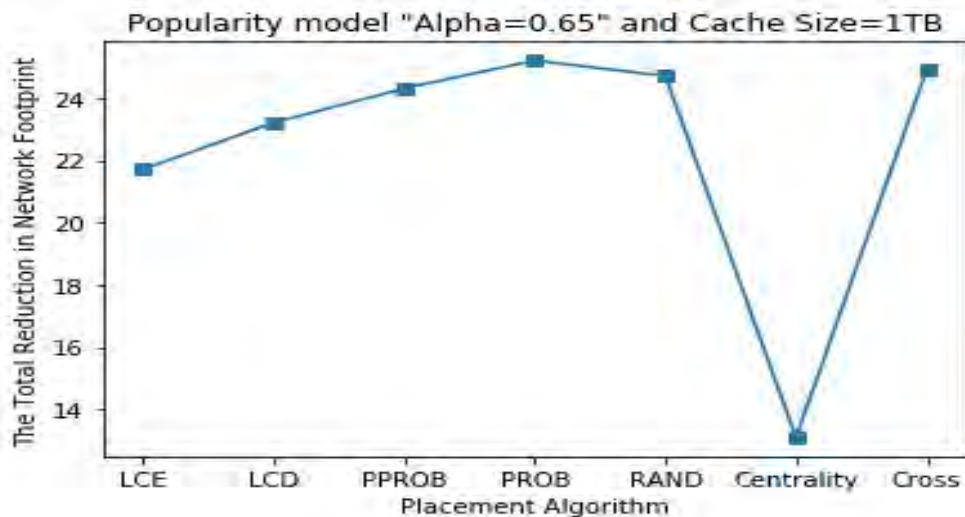


Figure 4.20. Server Load Reduction for Seven Placement Mechanisms Using 1 TB and  $\alpha = 0.65$

#### 4.4.3.5 Server Load Reduction Using 1 GB to 1 TB and $\alpha = 0.65$

The supplied diagrams in Figures 4.17 to 4.20 allow us to conclude that that for all seven mechanisms, the reduction in server load increases as cache size increases. The average in the Table confirms that the server load reduction of the LCD, PProb, and Prob mechanisms gave the best outcomes when examining the server load decrease as a performance parameter [16, 294], as shown in the Table 4.5.

However, PProb has the poorest reduction in server load performance at 1GB but is the greatest at 10GB and comes in second at 100GB. Finally, it is the best one in 1TB. Generally, the performance of reduction in the server load for all placement mechanisms gives increased performance when the cache size is increased.

**Table 4.5**

Server Load Reduction for Seven Placement Mechanisms Using 1 GB to 1 TB and  $\alpha = 0.65$

Cache Size	LCE	LCD	PProb	Prob	Rand	Centrality	Cross
1GB	5.2	7.1	3.4	4.7	4.5	3.6	5.9
10GB	9.8	9.1	10.6	10.6	8.3	7.9	9.2
100GB	20.7	29.7	28.5	28.1	26	16.2	23.2
1TB	43.4	50.2	52.4	51.3	49.9	25.8	47.3
Average	19.78	24.03	23.725	23.68	22.18	13.38	21.4

In addition, the PProb mechanism has the second level in its average server load reduction to four sizes. The other mechanisms, Prob, Rand, Cross, and LCE, gain the third and sixth levels, respectively. While the Centrality mechanism is the worst performance in the server load reduction, that is very clear when we see Figure 4.21.

The Rand mechanism successfully reduces server load because it deletes content for each request made to the server, replacing only one of the deleted items with the arrival of new content [266]. As a result, there is a significant increase in the decrease in server load. To give a potential field for node caching, it was further designed to copy the content file to a random node while the reference packet was being processed across the network.

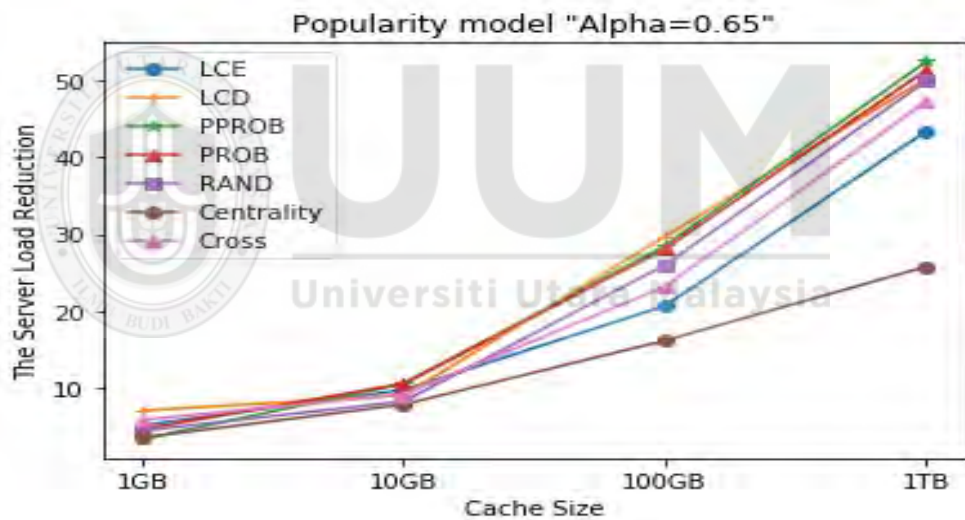


Figure 4.21. Server Load Reduction for Seven Placement Mechanisms Using 1 GB to 1 TB and  $\alpha = 0.65$

The results of the server load reduction of the PProb cache placement mechanism for various sizes are shown in Figure 4.21. It is evident that as the cache size grows, the server load decreases proportionally.

#### 4.4.3.6 Server Load Reduction Using 1 GB to 1 TB and $\alpha = 2.0$

The Table 4.6 is explained in Figure 4.22 the server load reduction for seven mechanisms in four cache sizes. Therefore, it can conclude that the Centrality mechanism has the worst values in the 1GB to 1TB cache sizes. If focusing on the average results, the Prob and LCD mechanisms obtain the best mechanism over the average results of all cache sizes. The results show that the server load reduction performance metric increases when increasing the cache size. At the same time, when the Zipf distribution is increased from  $\alpha = 0.65$  to  $\alpha = 2.0$ , the total traffic reduction performance metric is also increased.

**Table 4.6**

The Server Load Reduction for Seven Placement Mechanisms Using 1 GB to 1 TB and  $\alpha = 2.0$

Cache Size	LCE	LCD	PProb	Prob	Rand	Centrality	Cross
1GB	10.6	11.5	8.2	9.8	10.7	8.2	11.2
10GB	19.3	20.1	19.6	21.8	19.7	17.4	18.9
100GB	41.4	49.9	46.1	49	47.2	33.8	44.5
1TB	67.2	72.6	77.3	73.1	71.4	49.8	68.3
Average	34.63	38.53	37.8	38.43	37.25	27.3	35.73

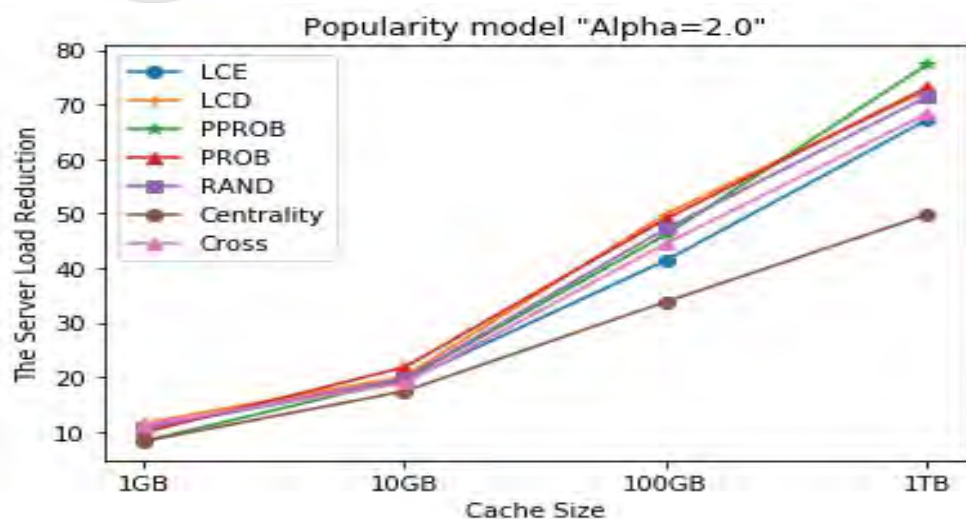


Figure 4.22. Server Load Reduction for Seven Placement Mechanisms Using 1 GB to 1 TB and  $\alpha = 2.0$

#### 4.4.4 Reduction in the Network Footprint

The total number of bytes saved through in-network caching measures the overall network footprint reduction. It was calculated using the Equation 3.5. Reduction the network footprint entails a reduction in the volume of data transmitted across the network to provide content to users. For VoD services, which deliver substantial quantities of video material, this efficiency is especially vital. A reduced network footprint generally results in diminished congestion, expedited delivery times, and enhanced utilisation of network resources, all of which enhance the QoS for end-users.

The following Subsections show the values of the network footprint reduction for each of the seven caching placement mechanisms for various cache sizes and Zipf distributions to assess the performance of the seven mechanisms chosen for VoD.

##### 4.4.4.1 Total Reduction in Network Footprint Using 1 GB and $\alpha = 0.65$

When the cache size is 1GB, Figure 4.23 shows the outcomes of lowering the network footprint for each of the seven placement mechanisms. The LCD method performs best in minimizing the network footprint 4.1. While other processes, such as LCE and Cross, have reduction values comparable to 2.7. Centrality and PProb methods provide the lowest network footprint reduction, respectively. This refers to the reason of these mechanisms are better utilizing local storage to keep traffic near the edge for small caches size [295].

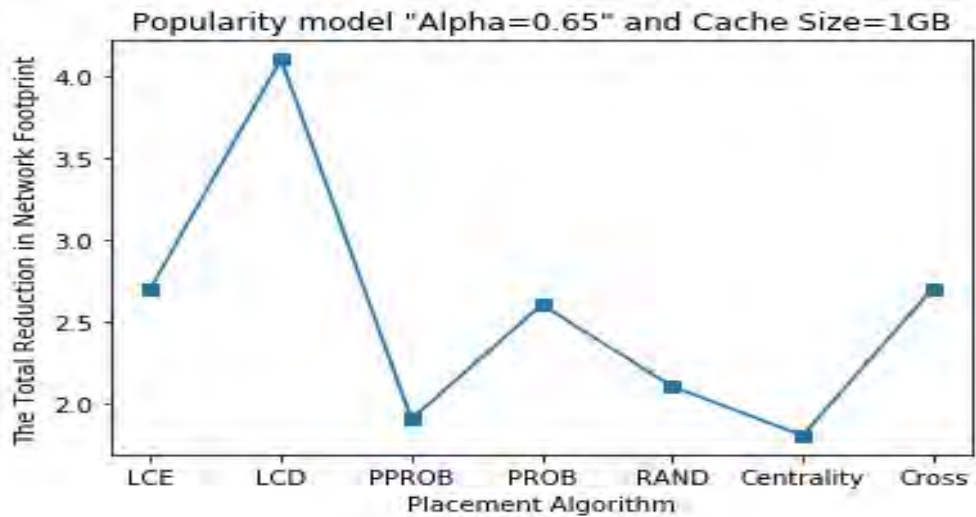


Figure 4.23. Total Reduction in Network Footprint for Seven Placement Mechanisms Using 1 GB and  $\alpha = 0.65$

#### 4.4.4.2 Total Reduction in Network Footprint Using 10 GB and $\alpha = 0.65$

When the cache size is 10GB, the network footprint reduction for all mechanisms is depicted in Figure 4.24. The LCD mechanism, which is more effective than the Rand method at reducing network footprint, again provides the maximum value. While the centrality approach once more provides the lowest value for reducing network footprint. Given that the cache capacity has been

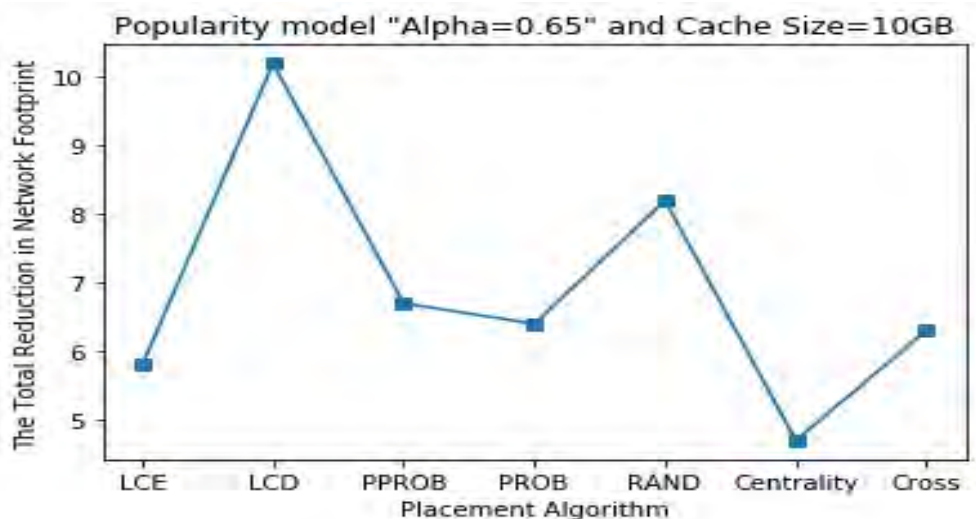


Figure 4.24. Total Reduction in Network Footprint for Seven Placement Mechanisms Using 10 GB and  $\alpha = 0.65$

#### 4.4.4.3 Total Reduction in Network Footprint Using 100 GB and $\alpha = 0.65$

Figure 4.25 shows the overall network footprint reduction for all mechanisms, with a cache size of 100GB and an alpha of 0.65. Once more, the LCD method reduces the network footprint the most, followed by Rand, Prob, Cross, PProb, and LCE, in that order. However, the Centrality mechanism experienced the least reduction in footprints due to its bottleneck. Thus, Centrality gets the worst result.

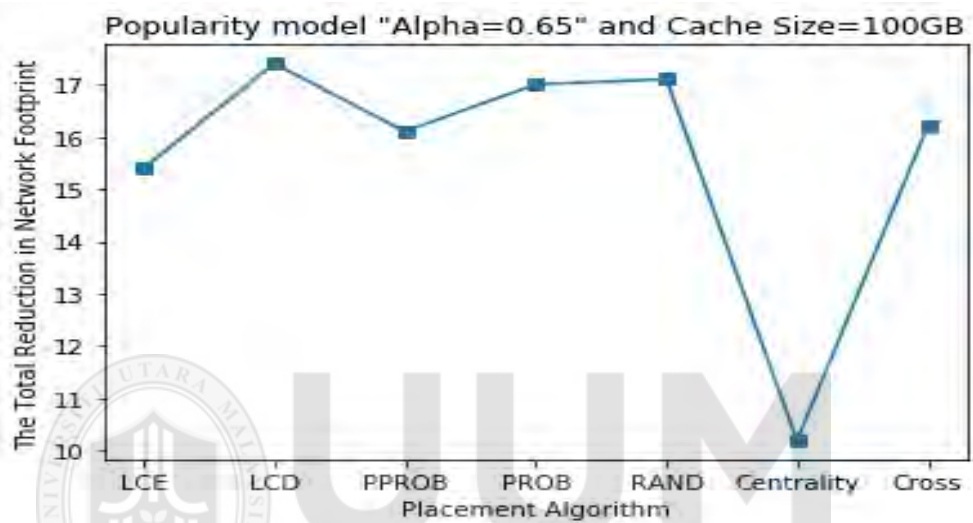


Figure 4.25. Total Reduction in Network Footprint for Seven Placement Mechanisms Using 100 GB and  $\alpha = 0.65$

#### 4.4.4.4 Total Reduction in Network Footprint Using 1 TB and $\alpha = 0.65$

The Figure 4.26 appears various attitudes toward the mechanisms. In this case, the Prob then Cross mechanism gives the highest value of the network footprint reduction. However, the Centrality mechanism again was the lowest due to misaligned content placement far from consumers [296]. In contrast, the results of the other four mechanisms (Rand PProb, LCD, and LCE) are in the middle, respectively.

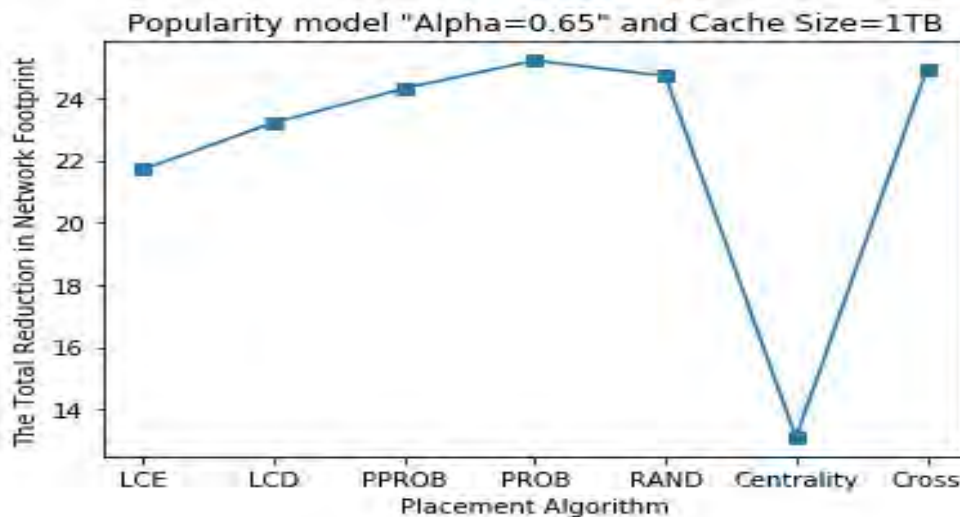


Figure 4.26. Total Reduction in Network Footprint for Seven Placement Mechanisms Using 1 TB and  $\alpha = 0.65$

#### 4.4.4.5 Total Reduction in the Network Footprint Using 1 GB to 1 TB and $\alpha = 0.65$

As a result, we may infer from Figures 4.23 to 4.26 that for all seven of the given mechanisms, the overall reduction in the network footprint increases as cache size grows. The LCD and Rand methods reduce the network footprint, while the Centrality mechanism yields the poorest result (7.45). However, Table 4.8 contains the averages of the four results for each mechanism.

In addition, from Table 4.7, we notice that Centrality and PProb are the worst in 1GB cache size. However, PProb has the middle-performance level of the total reduction in the network footprint in 10GB and 100GB cache sizes. Moreover, Prob has the first performance level when the cache size is 1TB. Therefore, we can conclude that the best level value of the total reduction in network footprint for the PProb mechanism is when the cache size is larger.

**Table 4.7**

Total Reduction in Network Footprint for Seven Placement Mechanisms Using 1 GB to 1 TB and  $\alpha = 0.65$

Cache Size	LCE	LCD	PProb	Prob	Rand	Centrality	Cross
1GB	2.7	4.1	1.9	2.6	2.1	1.8	2.7
10GB	5.8	10.2	6.7	6.4	8.2	4.7	6.3
100GB	15.4	17.4	16.1	17	17.1	10.2	16.2
1TB	21.7	23.2	24.3	25.2	24.7	13.1	24.9
Average	11.4	13.725	12.25	12.8	13.02	7.45	12.53

Furthermore, depending on the average performance results, we notice that LCD is the best, while Rand, Cross, Prob, Prob, and LCE mechanisms arrange sequentially in the middle. Therefore, the LCD mechanism offers the highest value for footprint reduction compared to the LCE and the Prob mechanisms, as illustrated in Figure 4.27 from the study [72, 297]. In contrast, the Centrality mechanism is the worst because it increases the likelihood of cache misses in a network [159] and causes the content node to disappear in the case of a high number of content delivery channels.

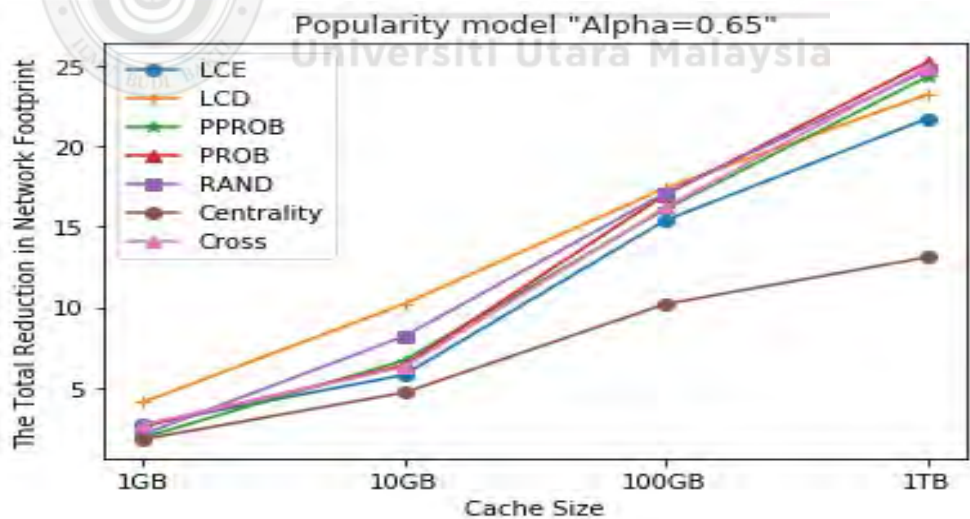


Figure 4.27. Total Reduction in Network Footprint for Seven Placement Mechanisms Using 1 GB to 1 TB and  $\alpha = 0.65$

The Figure 4.27 depicts the results for different sizes from 1GB to 1TB. For example, the total traffic reduction in the Centrality mechanism is 1.8 for 1GB cache sizes, for

10GB is 6.7, for 100GB is 10.2, and for 1TB is 25. Therefore, the conclusion is that when the cache size increases, the total reduction in the network footprint also increases. For example, the result of Centrality is 1.8 in 1GB cache size, and the result of Prob is 25.2 in 1TB.

#### 4.4.4.6 Total Reduction in the Network Footprint Using 1 GB to 1 TB and $\alpha = 2.0$

The Table 4.3 is explained in Figure 4.10 the total traffic reduction for seven mechanisms in four cache sizes. Therefore, it can conclude that the Centrality mechanism has the worst total traffic reduction values in the 1GB to 1TB cache sizes. If focusing on the average results, the Prob and LCD mechanisms obtain the best mechanism over the average results of all cache sizes. The results show that the total traffic reduction performance metric increases when cache size increases. With larger caches, repeated content requests (common in VoD) are increasingly satisfied in-network. This prevents duplicate requests from being sent upstream multiple times, particularly for popular video segments. This is especially relevant for VoD, where a small number of videos account for a large portion of the traffic (Zipf-like behavior) [295]. At the same time, when the Zipf distribution is increased from  $\alpha = 0.65$  to  $\alpha = 2.0$ , the total traffic reduction performance metric is also increased.

**Table 4.8**

Total Reduction in Network Footprint for Seven Placement Mechanisms Using 1 GB to 1 TB and  $\alpha = 2.0$

Cache Size	LCE	LCD	PProb	Prob	Rand	Centrality	Cross
<b>1GB</b>	3.6	5.2	3.2	4.1	3.5	2.9	4
<b>10GB</b>	7.1	11.5	8.1	7.9	9.6	6.1	7.8
<b>100GB</b>	16.8	18.6	17.6	18.3	18.5	11.5	17.3
<b>1TB</b>	23.1	23.8	27.3	26.7	25.9	14	25.6
<b>Average</b>	12.65	14.78	14.05	14.25	14.38	8.63	13.68

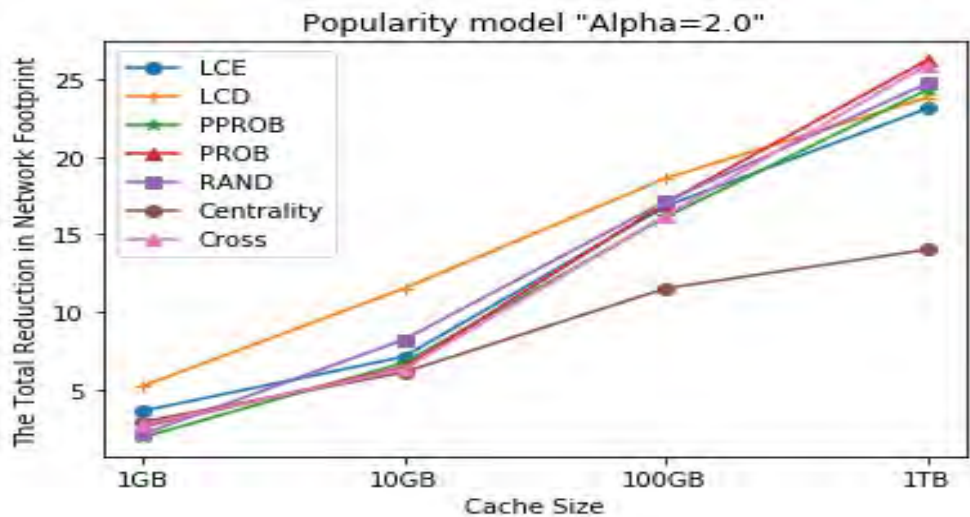


Figure 4.28. Total Reduction in Network Footprint for Seven Placement Mechanisms Using 1 GB to 1 TB and  $\alpha = 2.0$

#### 4.4.5 Functional Requirements of VoD in NDN

There are many challenges of caching video in NDN due to its large scale. Thus, specific requirements are needed to solve these challenges of caching VoD in NDN by linking them to the reviewed literature and then used in the proposed CPVP mechanism. The functional requirements are:

1. Minimising hop distance by caching near the consumer in the proposed CPVP mechanism to enhance the QoS for the end-users, resulting in a total reduction in network footprint.
2. Minimising the latency time that results in the average of the E2EDelay.
3. Avoid redundancy to increase the cache diversity by caching once in CPVP.
4. Enhancing the memory utilisation by increasing diversity in CPVP that results in increasing the average hit ratio and server load reduction due to the limited cache size and large scale of video.
5. Increasing diversity by dividing the video into three parts and caching each part in the router in CPVP.

These functional requirements are used to design the new explicit Coordinated Cache Placement mechanism via Video Partitioning (CPVP) in NDN during VoD services, as will be explained in the next chapter.

#### 4.5 Summary

In this work, the seven cache placement mechanisms are evaluated (LCE, LCD, PProb, Prob, Rand Cross, and Centrality) using different cache sizes (1GB, 10GB, 100GB, and 1TB) during VoD workloads to provide a deep understanding of the features of these NDN cache placement mechanisms by performing an empirical evaluation.

According to the results, it can be concluded that the Centrality placement mechanism is the worst one in all performance metrics, which are average hit ratio, average E2EDelay, server load reduction, and total reduction in network footprint. The Prob mechanism got the best time delay results, and the delay time was lowest by the PProb and LCD mechanisms. The LCD, Rand, and Prob mechanisms achieved the highest average hit ratio among the three mechanisms in sequential order.

Furthermore, the server load reduction performance metric is given the highest value in LCD, Prob, and PProb placement mechanisms. At the same time, the results of Prob and PProb mechanisms are the best in terms of total network footprint reduction.

Additionally, increasing the cache yields the highest results for all seven mechanisms. At the same time, the results are enhanced for all seven placement mechanisms when the Zipf distribution is increased from  $\alpha = 0.65$  to  $\alpha = 2.0$ .

In general, it can be said that the Prob method had the smallest time delay and provided adequate results for the performance criteria that were looked at. It yields the greatest results for the typical cache hit ratio, overall network footprint reduction, and server

load reduction for various cache sizes. These results make it a big challenge to enhance these performance metrics by designing new mechanisms. Therefore, the first objective is achieved in this chapter.

Overall, the Prob method demonstrated the smallest time delay and yielded satisfactory results for the examined performance criteria. It yields the greatest results for the typical cache hit ratio, overall network footprint reduction, and server load reduction for various cache sizes.

These results make it a big challenge to enhance these performance metrics by using the functional requirement to improve the placement caching mechanism in NDN during VoD services. This enhancement is achieved by designing new mechanisms. Consequently, this chapter accomplishes the first goal. The functional requirements are minimising hop distance by caching near the consumer to enhance the QoS for the end users in order to minimise average E2EDelay. The other requirement is to avoid redundancy to increase the cache diversity, which leads to enhancing the memory utilisation, the average hit ratio, and server load reduction due to the limited cache size and large scale of video.

These functional requirements are used to design the new explicit Coordinated Cache Placement mechanism via Video Partitioning (CPVP) in NDN during VoD services, as will be explained in the next chapter.

# CHAPTER FIVE

## DESIGN COORDINATION CACHING PLACEMENT BY VIDEO PARTITIONING (CPVP)

### 5.1 Introduction

Video on Demand (VoD) is a significant contributor to global network load, with streaming services like Netflix, YouTube, and Amazon Video enabling users to watch videos from various sources at any time and on various devices. With an expanding user base, many people under the age of 32 have grown up using VoD services without regular cable subscriptions [298]. Future systems (NDN) must run Content Delivery Networks (CDNs) geographically dispersed to make VoD streaming scalable. However, VoD's high daytime usage and adaptation to user demand have a negative impact on CDN performance, resulting in costly over-provisioning, traffic spikes, and reactive content eviction [299, 15, 300]. To meet future streaming VoD estimates of hundreds of millions of users, new mechanisms are required to improve content distribution effectiveness. However, according to content caching, it is simultaneously a challenge for NDN to meet QoS requirements to solve the problem of guaranteeing services in NDN and network delay, besides improving the overall performance of content delivery.

This chapter comes up with a new way to place caches by moving on from independent caching placements like LCE and implicit coordinated caching like LCD and Prob. It then suggests the explicit coordinated caching placement mechanism named Coordination Caching Placement by Video Partitioning mechanism (CPVP) to cut down on cache duplication and improve the network's content distribution diversity. To this end, a structure of NDN in the new placement caching mechanism was introduced that allows neighbouring CRs to cooperate to prevent redundant caching and to satisfy each other's requests that can't be met locally. In Chapter Six, we

used analytical modelling and simulation tests to evaluate the effectiveness of the coordinated caching mechanism. It greatly reduced cache redundancy and raised the cache hit ratio.

This thesis aims to design the CPVP by identifying three schemes that have the potential to address the future network for VoD content in NDN while also providing theoretical proofs. Our first scheme is the design of the Video Aggregation Scheme (VAS), which explains how the network aggregates the content of interest video to minimise the hop distance. In addition, it labels all the routers in the path. Our other two schemes use this labelling. This focus will benefit VoD providers, such as YouTube. Our second scheme involves the creation of a Video Caching Decision Scheme (VCDS), which employs an explicit partitioning placement mechanism on the path. This mechanism relates to whether the router caches the videos or not, and where each router maintains a copy. It adopts caching the three blocks of video chunks in three different routers on the path nearest to the consumer (the edge router and its two neighbours on the path). Our third scheme concentrates on the in-network caching of the Video Partitioning Scheme (VPS). We introduce a design of VPS that partitions large-scale video content into three blocks of chunks and distributes them across three routers. The term "block" denotes the multi-chunks stored in CS for identical video content, while "chunk" or "segment" denotes the portion of the video content that travels on the path to reach consumers. However, each video segment should contain multiple chunks according to a scalable video coding dataset (DASH/SVC dataset). Thus, the routers cache one-third of each video request, contributing to the overall cache efficiency by minimising memory consumption. Therefore, these three schemes will design CPVP to minimise the average delay. Simultaneously, it will result in a significant increase in both the overall network footprint reduction and the average hit ratio.

This chapter introduces a novel mechanism, known as the CPVP mechanism, for use in NDN during VoD services. Section 5.2 illustrates the NDN network model. Section 5.3 conducts a theoretical analysis of the problem, followed by a determination of whether coordination is necessary for the best placement caching or not. Section 5.4 explains the design of the CPVP mechanism, which requires creating four cooperative tables and designing three mechanisms: VAS, VCDS, and VPS. Section 5.5 then demonstrates the process of verifying and validating the newly designed mechanism. Finally, this chapter ends with the summary in Section 5.6.

## 5.2 Network Model

The NDN cache placement mechanism can benefit from a hypergraph's characteristics. Hypergraphs establish a relationship between the edge nodes and the midpoint nodes. However, each hyperedge can connect more than two nodes, modeling the multicast nature of NDN communication, where a content request might involve multiple routers and nodes. Therefore, determining the cache path capacity is the first step in this research. We consider a network  $N$  using hypergraph  $G = (V, E)$  to model the NDN network, where  $V$  is the set of nodes (e.g., router, server, and user) and  $E$  is the set of connections between  $V$  in NDN called edges or hyperedges. Whereas,  $V = \{v_1, v_2, \dots, v_n\}$  is the subset of CR (i.e., having the capacity of caching). Whereas the set of edges is  $E = \{e_1, e_2, \dots, e_m\}$  with  $e_i \in E$  for  $i = 1, 2, \dots, m$ , is the set of hyperedges. The other type of graph is the standard or simple graph in which each edge can only connect a maximum of two nodes. Obviously, the graph is the standard graph when  $|e_i| = 2, i = 1, \dots, m$ .

While the size of a standard graph is uniquely determined by  $n$  and  $m$ , the size of a hypergraph also depends on the number of its hyperedges; we define the scope of  $G$  as the sum of the number of elements in its edges:

$$size(G) = \sum_{e_i \in E} |E_i| \quad (5.1)$$

It is important to note that there is a one-to-one correspondence between hypergraphs and Boolean matrices. Indeed, a hypergraph's incidence matrix refers to any  $n * m$  matrix  $A = [a_{ij}]$  such that  $a_{ij} \in \{0, 1\}$ , where each column  $j$  with a hyperedge  $e_j$  and each row  $i$  is associated with a vertex  $v_i$  in a hypergraph  $G$ .

A hyper-arc is an ordered pair of (probably empty) disjoint subsets of vertices,  $E = (X, Y)$ ; whereas,  $X$  is the end of  $E$  while  $Y$  is its start. In the following, the  $X$  and the  $Y$  of hyper-arc  $E$  are represented by  $T(E)$  and  $H(E)$ , respectively. So, assume that  $N = (V, E)$  is a hypergraph while  $t$  and  $s$  are two distinct nodes, the sink and the source, respectively.

The partition of  $V$  is a cut  $T_{st} = (V_s, V_t)$  where  $V_s$  and  $V_t$  are two subsets so that  $t \in V_t$  and  $s \in V_s$ . The cut given  $T_{st}$ , its  $E_{st}$  cut-set is the subset of all  $E$  hyper-arcs so that  $H(E) \subseteq V_t$  and  $T(E) \subseteq V_s$ . Remember that,  $t$  is no need to disjoint from  $s$  by deleting the hyper-arcs of a cut-set [89]. Here, we assume that the subset contains three edges of neighbour routers; see [301, 82, 302, 303, 304]. So, we assume the sub-set is  $E_{st} = \{v_1, v_2, v_3\}$  on the path to the source  $V_s$ .

Using a popular network, the Tree network architecture in NDN, the study's suggested hypergraph model is demonstrated. Thus, the new model description is defined by hypergraph theory as shown below:

$T$  is a network topology which includes a set of  $E$  edges and  $V$  vertices as:

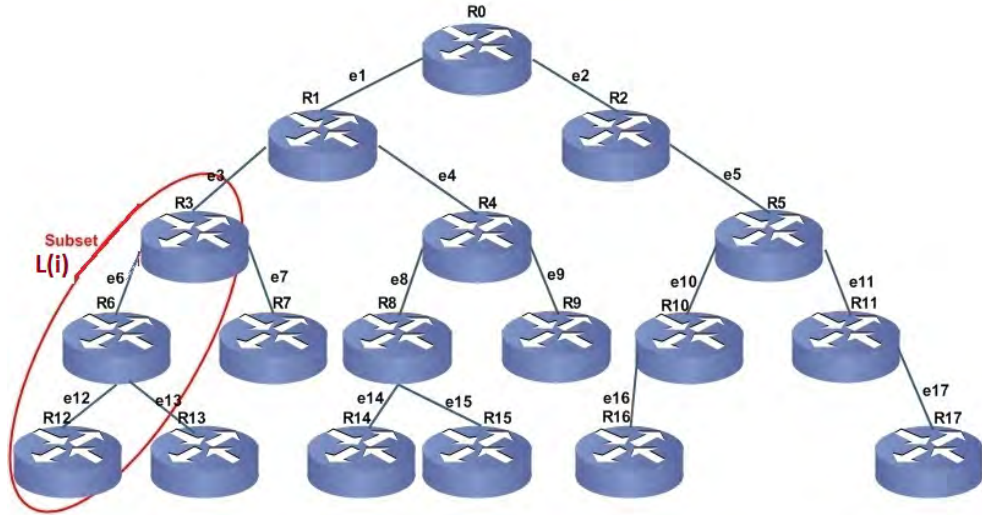


Figure 5.1. Tree Topology

$$T = (V, E) \quad (5.2)$$

Such that Tree topology supports the NDN formula as:

$$T' = (V, E) \quad (5.3)$$

Whereas  $V$  may be a router server and user, if we assume that  $V$  is a set of routers,  $R$ . Thus,  $V = R_0, R_1, R_2, \dots, R_{16}, R_{17}$  and  $E = e_1, e_2, e_3, \dots, e_{16}, e_{17}$ . By the Internet homogeneous, each edge contains a significant pair interconnection as  $e_i$ . This means that  $e_1 = R_0, R_1, e_2 = R_0, R_2, e_3 = R_1, R_3, e_4 = R_1, R_4, e_5 = R_2, R_5, e_6 = R_3, R_6, \dots, e_{17} = R_{11}, R_{17}$ , while  $T_{st} = \{R_{12}, R_6, R_3\}$  is an example of a subset if the consumer asks a request from the router  $R_{12}$  as shown in Figure 5.1.

For example, if a consumer initially requests video from  $R_{12}$  but doesn't locate it until reaching  $R_0$ , the video is then sent back to the consumer and cached in three different blocks within Subset  $L(i)$ . If a consumer requests a previously viewed video from  $R_{13}$

and finds it within one hop distance, the video is returned to the consumer without being cached again.

In NDN, the measured latency rate between two content routers (CRs)  $R_i$  and  $R_j$  can be determined by determining the static distance  $d_{ij}$  between them. The routers' capacity in that direction can also influence the latency rate. It is anticipated that  $k - 1$  CRs in  $V$  closest to the CR  $R_i$  will cooperate with it, with "closest" referring to the smallest distance. Our goal is to avoid caching the same chunks in these CRs. Thus, in the specific case of VoD service delivery, we use latency as a primary parameter to determine the distance length, ensuring smooth video streaming and minimal buffering. For a variety of reasons, we use latency as a primary parameter:

1. We hope to reduce the delay. Video chunks that are not under the CR's responsibility  $R_i$  must be collected from cooperating  $R_i$  neighboring CRs. Therefore, the RTT from  $R_i$  to any of its cooperating CRs should be minimised to ensure video continuity.
2. We don't care about bandwidth issues. The distance must take into account the capacity of the connections because bandwidth in conventional video services is an essential constraint. Nevertheless, most of the servers are subject to this constraint. The distributed and multicast nature of NDN, coupled with the high performance of the new generation of CRs, would overcome the bandwidth limitations. Therefore, when considering distance, we disregard the CR bandwidth and instead validate the latency constraint.

However,  $L(i)$  is the sub-set of closest CRs to  $V$ . Hence,  $L^+[i]$  is the set of  $L(i) \cup \{R_i\}$ .

Initially, every CR should define its label  $L$ . Our goal is for each CR to be as close

to all the various  $L(i)$  as possible, not just its labell. By  $L(i)$ , we note that of the  $k - 1$  CRs have the other  $k - 1$  labell which is collectively the closest to  $R_i$ . We use this labelling in the design of the proposed CPVP cooperative caching placement mechanism, particularly in VAS. We labell each router on the path from the consumer until the data is received. This allows us to identify the three nearest routers to cache with the proposed VPS if VCDS decides to cache the video, as we will explain in detail in Section 5.4.

The total distance in a hypergraph network can be defined as the sum of all pairwise distances between nodes within the hyperedges of the hypergraph. The total distance can represent the efficiency or cost of data transfer across a network involving multiple interacting nodes such as capacity, hop distance, and latency. The total of distances  $d_i$  between an offered CR  $R_i$  and the CRs in  $L(i)$  is referred to as the  $R_i$  rainbow distance, while  $d_i$  is calculated as in Equation 5.4:

$$d_i = \sum_{r_j \in L[i]} d_{ij} \quad (5.4)$$

It is NP-hard to determine the optimal labell assignment, i.e., the goal is to minimise the sum of all rainbow distances [305]. The rainbow distance, on the other hand, entails identifying the best route across numerous hyperedges, acting as a theoretical expansion rooted in the variety of paths and distances found in hypergraphs.

To prove our CPVP mechanism's performance rate, we start with the lower bound description. Because of an instance of the problem, a lower bound solution can determine by setting that each CR with its nearest CRs collectively store the  $k$  labells, formally  $L(i) = N(i)$  for each CR  $R_i$ .

In NDN, fractional distance could represent the distribution of retrieval costs across different paths to fetch a data chunk, accounting for cache proximity, path diversity, and network congestion. In such a case, the fractional distance might be used to indicate how a request for a data chunk is split across multiple cache locations or retrieval paths to optimize delivery. Thus, we name the fractional distance, referred to as  $\bar{d}_i$ , the sum of the length of the distances between a CR  $R_i$  and its nearest CR neighbors, so

$$\bar{d}_i = \sum_{r_j \in N[i]} d_{ij} \quad (5.5)$$

This research indicates that the proposed CPVP mechanism can more effectively use Hypergraph to achieve the goals set out in **Chapter One**. The NDN cache placement mechanism can utilise hypergraph attributes. The study aims to improve where NDN caches are placed by looking at the cache path capacity of hypergraphs and showing how edge and the midpoint nodes interact with each other. In coordinated caching techniques, the caching strategy entails collaboration across numerous nodes to enhance data placement. Hypergraphs can effectively express the relationships between nodes and the data they cache, thereby modeling this cooperative behavior. In VoD, a hypergraph can depict each hyperedge as a collection of nodes working together to cache a particular piece of content due to minimise the cost of data transfer across a network involving multiple interacting nodes such as capacity, hop distance, and latency.

### 5.3 Theoretical Analysis

This section delves into the analysis of the content placement problem, concluding that the implementation of new caching placement is crucial. Next, we will examine

both coordination and non-coordination placement strategies to determine the most effective approach for the VoD workload in NDN.

In this section, we formulate and perform an analysis of a content placement issue. The content placement issue pertains to the distribution of video chunks to the cache nodes, with the aim of minimizing the cost of caching placement, such as hop distance and capacity. Whereas the distance is discussed in Section 5.2 to be minimised using hypergraph and cooperated caching placement to avoid content redundancy and inefficient routing paths. However, the simultaneous handling of multiple online video requests can strain the network's performance, particularly impacting caching placement mechanisms for high VoD workload periods. Content redundancy and inefficient routing paths make this stress even worse, leading to long End-to-End Delay (E2EDelay), heavier server loads, and even sudden network interruptions [71]. In VoD, the main reasons for these problems are limited cache space and large video scale that will be proved in Subsection 5.3.1.

However, the important part of the problem is that the simultaneous handling of multiple online video requests can strain the network's performance. This part is represented by  $\lambda_{c,i}$ , which is the average request ratio for chunk  $c$  at router  $R_i$ . The subsection 5.3.2 proves the effect of the average request rate and analyses the coordinated and non-coordinated caching placement based on the request rate to find the better one. Moreover, we use chunks because using chunks for video transfer is often considered better than traditional segmentation because chunking helps to minimise latency and optimize bandwidth usage, leading to smoother streaming experiences, especially in adaptive streaming protocols. Table 5.1 contains some important quotations related to the proposed model, which can facilitate a smooth discussion.

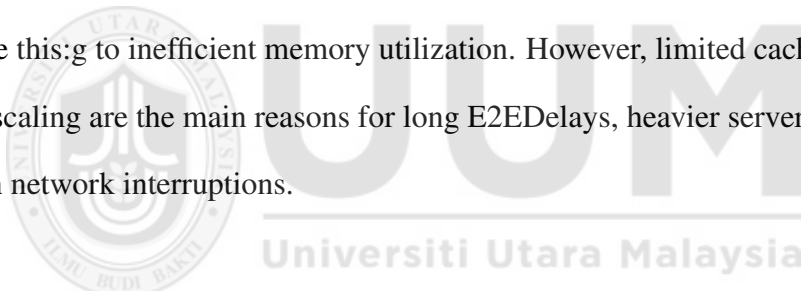
**Table 5.1**  
Related Parameters in Model

Notation	Definition
$V$	The cache nodes set.
$n_c$	original server that collects all chunks in $V$ .
$R_i$	router that has a limited $C(i)$ cache capacity.
$\lambda_{c,i}$	The average of request ratio for chunk $c$ at every router $R_i$ .
$P_c$	The percentage $p$ of request for chunk $c$ .
$\lambda_{c,i} * P_c$	The frequent request of chunk $c$ at router $R_i$ .
$R_c$	The router that request chunk $c$ .
$R_i$	The intermediate router deliver request message.
$d_{c,i}$	The hop distance reduction to cache the chunk $c$ at router $R_i$ for request chunk $R_c$ .
$b_{c,i}$	The binary variable show whether cache chunk $c$ on router $R_i$ .
$C$	The total number of chunks.
$C(i)$	The maximum cache capacity of router $R_i$ , taking chunk as the unit.
$n_i$	The node $i$ that caches the chunk $c$ .
$\lambda^{\setminus}$	The cache miss rate in coordinated routers.
$\lambda$	The cache miss rate of a single cache $i$ without coordination.
$K$	The number of coordinated routers in $V$ .
$\tau_i$	The average object residence time in the cache $i$ .
$\tau_i^{\setminus}$	The average object residence time in the coordinated cache $i$ .
$\lambda_i$	The request rate of a single cache $i$ .
$\lambda_i^0$	The cache miss ratio of chunk $i$ is
$k * C$	Coordinated group of cache size.
$k * \lambda_i$	The delivery rate of user request at a single cache.

### 5.3.1 Analysis of Modeling Content Placement Problem

In this section, we formulate and perform an analysis of a content placement issue. The content placement issue pertains to the distribution of chunks to the cache nodes, with the aim of minimizing the cost of caching placement, such as hop distance and capacity. Content redundancy is one important reason for making the problem even worse, leadinIn NDN, customers demand data by submitting an Interest that explicitly addresses a content file's Data chunk. Each NDN router contains a content store (CS) to cache Data. The caching placement mechanism determines if and where to cache the video chunk. By default, NDN caches data as LCE on every router visited by the

data [306]. Thus leads to redundancy that wastes the caching capacity. A potential solution is to selectively cache content only on the three-edged router and partition this video to these three routers. Whereas, the three-edged router refers to the edge router and its two neighbor routers on the path. This solution is to reduce data redundancy, memory length, the lag and loss data packet. This also good to increase memory usage, diversity which leads to increase average hit ratio, and E2EDelay. It has been impractical to achieve this solution with file popularity because of multi reasons. These reasons are involving to rate a large number of files when new data inserted, increment in demand, in addition to a large VoD file. Therefore, partitioning out video files at the edge, whereas, our design of placement mechanism spreads a file over multiple routers along the three edge's path from the client to the source, with the head nearer the client and the tail nearer the source. There are multi functional requirements to achieve this: to inefficient memory utilization. However, limited cache size and large video scaling are the main reasons for long E2EDelays, heavier server loads, and even sudden network interruptions.



Each node can cache or not the passed data, whereas the server must cache it all. Simultaneously, the servers and users exert influence on the nearby cache nodes. The system transmits all videos in chunks, with each chunk having the same size as the basic cache unit. A server  $n_c$  is an original database that collects all chunks in  $V$ . The forwards of the relevant video rely on the forwarding entry of FIB, which is established through specific routing mechanisms. Each video is delivered in multiple chunks. Therefore, calculate the request rate for each interest chunk instead of the request rate for interested video. Each  $R_i$  is a router that has a limited  $C(i)$  cache capacity. Furthermore, assuming that every  $R_i$  node delivers request messages (Interest chunk) at a rate of  $\lambda_{c,i}$  with a percentage of  $P_c$  of requesting chunk  $c$ . Therefore,  $\lambda_{c,i} * P_c$  demonstrates the frequent request of chunk  $c$  at router  $R_i$ .

Here, we introduce the concept of content placement with optimal memory utilization, illustrating how cached content reduces path length. For instance, chunk  $c$  is requested from router  $R_c$  and forwarded to its original server  $n_c$ . Hence, path  $(R_c, n_c) = R_c, \dots, n_c$  is the forwarding path of the request between  $R_c$  and  $n_c$ . If an intermediate cache router  $R_i$  is satisfied with the request message, then the path will be decreased to  $R_c, \dots, R_i$ . Thus, the reduced length  $R_i, \dots, n_c$  is demonstrated as the benefit  $d_{c,i}$  of content placement. The path length value equals the hop count from the caching router to the server on the distribution path. Note that,  $b_{c,i}$  is a positive integer if  $R_i$  is on the path  $(R_c, n_c)$ . Apparently, if  $R_i$  not in the path  $(R_c, n_c)$ , then  $b_{c,i}$  equals to zero.

The maximization problem commonly found in optimization scenarios such as network resource allocation, content caching, or traffic management in NDN as follows:



$$\text{Max} \sum_{i \in V} \sum_{c \in C} \lambda_{c,i} \cdot P_c \cdot d_{c,i} \cdot b_{c,i} \quad (5.6)$$

$$c.t \sum_{c \in C} P_c = 1 \quad (5.7)$$

$$b_{c,i} = 0/1 \quad (5.8)$$

$$\sum_{c \in C} b_{c,i} \leq C(i) \quad (5.9)$$

This optimization model is likely trying to maximise network performance by adjusting content placement, bandwidth allocation, or routing strategies. In many network or caching scenarios, the goal is to reduce latency, minimise cost, or optimize bandwidth while ensuring that content is delivered to users or nodes efficiently. Where  $n_i$  is the node  $i$  that caches the chunk  $c$  with the average request rate of this chunk  $\lambda_{c,i}$ .  $P_c$  refers to the probability of chunk  $c$  to be available, which reflects the popularity of the chunk. The availability of chunks refers to the accessibility of data packets (called "chunks") that are named and cached across the network. NDN focuses on content-centric communication, where data is requested by name, and once retrieved, it is cached at various network nodes, allowing future requests for the same content to be served from nearby caches instead of the original server. Frequently requested chunks are more likely to be cached across multiple nodes, improving their availability. Popularity-based caching strategies can ensure that commonly accessed content remains readily available in different parts of the network. While  $d_{c,i}$  refers to the benefit of placement caching chunk  $c$  at node  $R_i$  for the request from node  $R_c$ . While  $b_{c,i}$  denotes a binary variable which equals to 1 if the chunk  $c$  caches at node  $R_i$ . In conclusion from Equation 5.9, the capacity is a constraint for each cache node; hence, the proposed cooperative caching is needed.

### 5.3.2 Analysis of Coordinated and non-coordinated Caching Mechanism

The performance of the proposed cooperative caching approach is now analyzed based on a theoretical perspective. Overall, the performance metric is the rate of a cache miss. In this subsection, we focus on the miss rate instead of the hit rate which is more informative and critical, due to the importance of optimizing system performance or identifying inefficiencies. In a caching system, for example, a cache hit means data was successfully retrieved, while a miss triggers additional latency and resource use. Understanding the miss rate helps system administrators focus on reducing delays and improving performance. Therefore, the performance of an excellent system gives little

cache miss ratio.

The simplified design of the proposed cooperative caching method in NDN was compared (represented in Figure 5.2) with two caching methods which are the parallel structure of  $k$  CRs (see Figure 5.3) and  $k$  CRs connect in a series circuit (see Figure 5.4).

The proposed cooperative caching design with  $k$  homogeneous caches working together in Figure 5.2 to theoretically analyze the caching performance of the collaborative caching architecture and non-cooperative caching architecture. However,  $K$  refers to the number of routers, while homogeneous means that each router get same size of CS. Based on this analysis, the significant characteristic time  $\tau_i$  identify of caching,  $\lambda$  is a function of processing the request arrival. In contrast,  $C$  is the cache size, as well as the pattern of request. There are two concepts for the design of hierarchical caching, which are categorized as parallel and series caching. The principles of design are then employed to manage the structure of the cooperative caching. The performance of the proposed structure design as coordinated caching establishes to be superior to the structures of traditional hierarchical non-cooperated caching, as shown in **Proposition 1 and Proposition 2**:

Every cache store separates chunks and creates a group of cooperated cache size  $k * C$ . The LRU replacement mechanism is used on every single cache to ensure that each group of CRs evicts its video file at the same time. Although the coordinated caching placement mechanism group is not precisely at the same time on each router in the group, thus, the delivery rate of the users' requests at the approximating single cache is  $k * \lambda_i$ . In conjunction with Che Y. et al [155], the cache miss rate created by the cooperative group group is defined as  $\lambda' = k * \lambda_i e^{-k \lambda_i \tau_i}$ . In cooperated systems, multiple caches work together to share and retrieve content, improving overall efficiency and

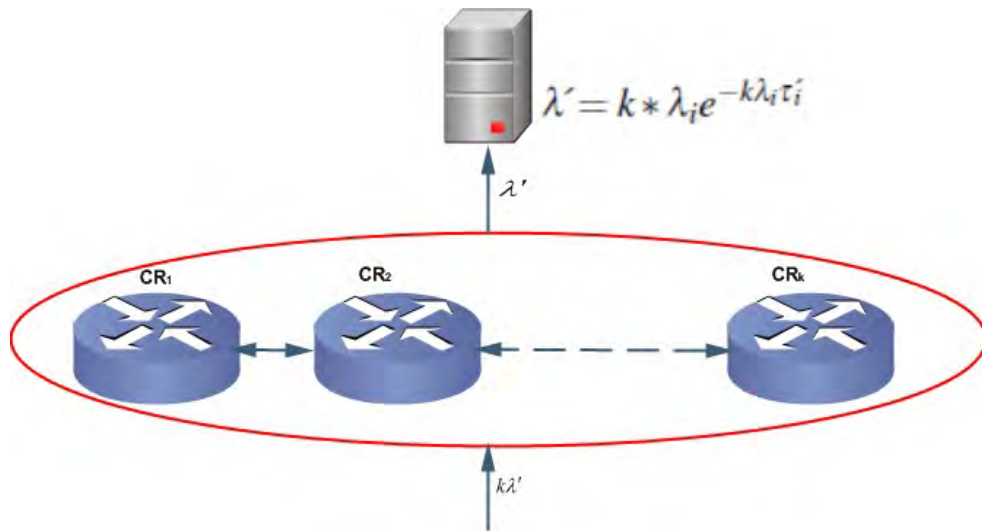


Figure 5.2. Coordinated Caching

reducing individual cache miss rates. Whereas,  $\lambda_i e^{-k\lambda_i\tau_i}$  reflects how cooperation among caches helps reduce the miss rate. The exponential decay factor,  $e^{-k\lambda_i\tau_i}$ , captures the impact of the cooperation. Thus, if the number of cooperated routers  $k$  increases, or if object residence time  $\tau_i$  increases, the miss rate decreases. For example, if we have 3 caches,  $k = 3$ , in a cooperative caching system, and the original miss rate of an individual cache is  $\lambda_i = 0.05$ , the coordinated caching system's miss rate  $\lambda'$  will be smaller than  $\lambda_i$  depending on the average object residence time  $\tau_i$ .

**The first structure is an individual parallel cache** as in Figure 5.3, homogeneous caches  $k$  of a cache size  $C$  are linked to a server. Every cache delivers a request of video chunk  $i$  with a request rate equal to  $\lambda_i$ . A cache miss occurs when chunk  $i$  is not in cache. The cache miss ratio of chunk  $i$  is represented as  $\lambda_i^0$ . Thus, the ratio of the total cache miss for the  $k$  caches is  $\lambda_m = k \cdot \lambda_i^0 = k \cdot \lambda_i e^{-\lambda_i\tau_i}$ , where  $\tau_i$  is the maximum interarrival time of the chunk  $i$  between two neighboring cache hits [307, 155, 308].

**Proposition 1:**

Coordinated caching achieves at least similar performances as the non-coordinated

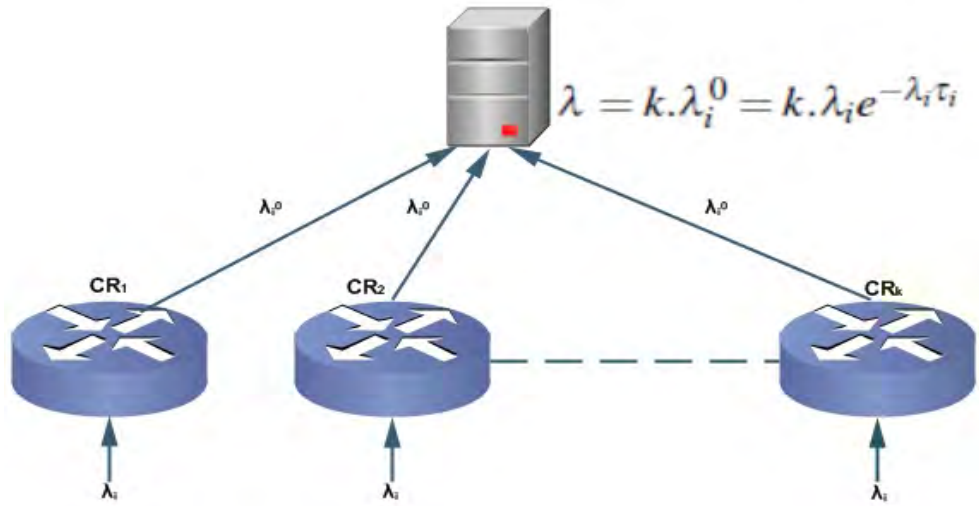


Figure 5.3. Non-coordinated Parallel Cache



Figure 5.4. Non-coordinated Cache in Series

parallel caches.

**Proof** The proof begins by acknowledging that the function  $f(\tau_i) = e^{-\lambda_i \tau_i}$  is both continuous and monotonically decreasing. In order to prove that  $\lambda \geq \lambda'$  for any  $k \in N$  and  $C \in N$ . The function  $u(\tau_i) = \sum_{j=1, j \neq i}^N f(\tau_j) = \sum_{j=1, j \neq i}^N e^{-\lambda_j \tau_j}$  is also continuous and steadily over time decreasing since  $u(\tau_i)$  is the sum of  $f(\tau_i)$ . According to [307, 155, 308],  $\tau_i$  and  $\tau'_i$  can be derived by the specific these Equations:

$$\sum_{j=1, j \neq i}^N (1 - e^{-\lambda_j \tau_i}) = C$$

$$\sum_{j=1, j \neq i}^N (1 - e^{-k \cdot \lambda_j \tau'_i}) = kC \quad (5.10)$$

Since  $C \in N$ , and  $k \in N$  then obtaining

$$\sum_{j=1, j \neq i}^N e^{-\lambda_j \tau_i} \geq \sum_{j=1, j \neq i}^N e^{-k \cdot \lambda_j \tau_i}$$

As  $u(\tau_i)$  is monotone reducing, it can be concluded that  $\tau_i \leq k\tau_i$ . Because  $f(\tau_i)$  is also monotone reducing, to know that:

$$e^{-\lambda_i \tau_i} \geq e^{-k\lambda_i \tau_i} \quad (5.11)$$

Multiply both sides of Equation (5.11) by  $k\lambda_i$ . The proof follows.

**The second structure is the non-coordinated cache in series** as demonstrated in Figure 5.4. For each cache, the request ratio  $\lambda$  from the customer  $i$  is still identical and denoted as  $\lambda_i$ . The missed stream forwards to the next-hop cache on the path to the server, instead of forwarding it straight to the server. The missed stream  $\lambda_{ki}^0$  of the cache  $k$  is the missed of the multi-cache stream system because only the  $k$ th cache is linked to the server. Therefore, discovering the expression of  $\lambda_{ki}^0$  is the convergence point. Whereas, a convergence point in a network is where multiple paths, data flows, or connections come together, playing a crucial role in routing, data distribution, and network efficiency. It is not easy to conclude  $\lambda_{ki}^0$ , even though the structure is simple, since the exact distribution function of the missing stream  $f_{ki}^0(t)$  includes an infinite number of terms [307, 155, 308]. Consequently, the precise miss ratio cannot be deduced due to computational complexity. However, the delivering request ratio at the  $k$ th cache is  $\lambda_{ki} = \iota \cdot \lambda_i$ , where  $\iota$  is a constant and  $1 \leq \iota \leq k$ . Let the miss rate be a function  $f_{ki}^0$  of  $\lambda_i$ , to have  $f_{ki}^0 = \lambda_{ki}^0 = \iota \lambda_i e^{-\iota \lambda_i \tau_{ki}}$ . Recall that the miss ratio of

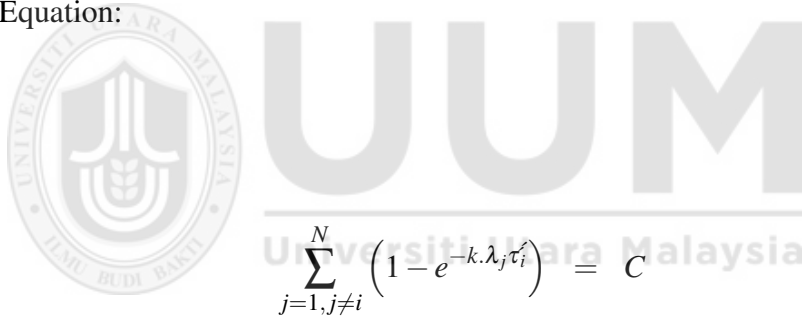
coordinated caching is  $f'_i = \lambda' = k * \lambda_i e^{-k\lambda_i\tau_i}$ .

**Proposition 2** Coordinated caching achieves, at most, a similar max miss rate as the non-coordinated caches in a series.

**Proof** In order to prove that, for any  $C \in N$ , and  $k \in N$  since have  $\max(f_{ki}^0) \geq \max(f'_i)$ . The value of  $\tau_{ki}$  can be delivered as follows:

$$\sum_{j=1, j \neq i}^N (1 - e^{-\lambda_j \tau_{ki}}) = C \quad (5.12)$$

Instead of directly comparing  $\tau_{ki}$  with  $\tau_i$ , another variable  $\tau'_i$  was used to setup the below Equation:



$$\sum_{j=1, j \neq i}^N (1 - e^{-\lambda_j \tau'_i}) = C \quad (5.13)$$

Since  $k \geq \iota$ , combining Eqs. 5.12 and 5.13, We have  $\tau_{ki} \leq \tau'_i$ . Applying the same method as that in the proof of 5.3.2 on Eqs. (5.13, 5.11), obtaining that  $\tau'_i \leq \tau_i \leq \tau_{ki}$ .

The first deviation of  $f_{ki}^0 = 0$  is:

$$\iota e^{-\iota\lambda_i\tau_{ki}} + \iota\lambda_i \cdot e^{-\iota\lambda_i\tau_{ki}} \cdot (-\iota\tau_{ki}) = 0 \quad (5.14)$$

then  $\lambda_i = (\iota\tau_{ki})^{-1}$ . The second derived of  $f_{ki}^0$  is less than zero, so

$$\max \binom{0}{f}_{ki} = (e \cdot \tau_{ki})^{-1} \quad (5.15)$$

similarly,

$$\max \binom{0}{f}_i = (e \cdot \tau_i)^{-1} \quad (5.16)$$

Since  $\tau_{ki} \leq \tau_i$ , then it can be conclude that

$$\max \binom{0}{f}_{ki} \geq \max \binom{0}{f}_i \quad (5.17)$$

Notice that the exponential value of  $f'_i$  reduces more quickly than  $f_{ki}^0$ , which indicates that the proposed coordination caching has at least similar performance for highly requested videos and better performance for medium requested videos. Therefore, cooperative placement caching is stronger than non-cooperated placement caching. To summarize, both the parallel and series non-coordinated models are less effective than the coordinated model. We expect the proposed design, which combines cooperated and series methods, to outperform the standard NDN and have a higher cache hit ratio. Therefore, we propose a mechanism that coordinates caching placement by dividing the video into multiple coordinated routers along the path.

#### 5.4 Coordinated Caching Placement by Video Partitioning (CPVP)

In NDN, customers demand data by submitting an interest that explicitly addresses a content file as chunks of data. The caching placement mechanism determines whether

and where to cache the video chunk. By default, NDN caches data using the LCE cache placement mechanism, which caches it on each router it visits. This mechanism results in redundancy, which wastes the caching capacity. So, the first goal of this study is to test and implement the seven cache placement mechanisms for VoD workload in terms of their average E2EDelay performance. This will help us meet the functional requirements listed in Chapter Four and come up with a solution.

A potential solution is to design the proposed CPVP mechanism that selectively caches content only on three routers and partitions this video's file to these three routers. One of these three routers is an edge router; the other two are neighborhood routers along the path. This section provides a detailed explanation of the CPVP mechanism. This solution reduces data redundancy, memory length, latency, and average hop distance. It also enhances memory usage and diversity, resulting in a higher average hit ratio, higher server load reduction, a higher total reduction in network footprint, and a reduction in E2EDelay. It has been impractical to achieve the solution of proposing CPVP to cache video based on its popularity for many reasons. These reasons include rating many files when new data is cached, increasing the demands, and a large VoD file. These factors are likely to exacerbate the issue rather than provide a solution. Therefore, our design of the placement mechanism CPVP spreads a file over three routers on the path from the client to the source, with the head nearer the client and the tail nearer the source. This chapter provides a detailed illustration the design of CPVP mechanism with its three schemes VAS, VCDS, and VPS, which satisfies multiple functional requirements to enhance the QoS for the end user, as follows:

1. Minimizing hop distance that result in the total reduction in network footprint:
  - a) Design the VAS that transfers entire video chunks from the server to the consumer or aggregates three blocks of video chunks from three

neighboring routers, then forwards them to the consumer. In addition to labelling all routers in the path with the number of the router sequence. For more details, see Section 5.4.2.

- b) Design VPS for more attention to precious cache space that is close to clients. The first block of the video at the edge stores closer to the consumers, while the second and third blocks store at the two neighbors on the path. This valuable resource of each router can share more files. Therefore, the decision of store or not and where to store is achieving by the proposed VCDS.

## 2. Minimizing latency time that result in the average of the E2EDelay:

- a) Utilising the basic NDN forwarding mechanism in our design because it considers as the best forwarding mechanism according to the study of Aloulou et al. [65] they stated that by comparing it with four forwarding mechanisms. That is because it offers the lowest throughput with minimum latency time.
- b) Using the Least Recently Used replacement mechanism (LRU). However, it bases on time and time is very important for a video request. Therefore, it deletes the oldest data, which leads to avoiding occupation and reducing latency.
- c) Design VCDS that aims to reduce transmission latency by gradually saturating the edge of a cache network through in-network caching.
- d) Design VPS, that cache the head chunks of video near to the consumer, because past studies [82, 309, 306, 310] indicate that users prefer to discontinue downloads. Hence, head chunks of video are more common than tail chunks. For more details, see Section 5.4.3.

## 3. Avoiding the data redundancy:

- a) Design the VCDS that decides to cache the same data one time if and only

if the hop distance  $X$  is more than three and the sequence of current router is less than four ( $X > 3 \& R < 4$ ) as explained later in details.

4. Enhance the stability of caching placement, leading to a higher cache hit ratio:
  - a) Design VCDS by using simple conditions ( $X > 3 \& R < 4$ ). However, this condition does not base on popularity to avoid complexity and occupied cache due to the large size of video.
5. Enhancing the memory utilization and increase diversity that result in increasing the average hit ratio and server load reduction:
  - a) Design VPS that partitions a file into three blocks. Each block of chunks is a cache at the three neighbor routers. For content such as video, the user-perceived latency based on how soon the application will have sufficient chunks to start. Retrieval delay for certain pieces will mask the amount of time it takes for the device to retrieve the head [311]. Hence, this scheme increases the average hit ratio.
  - b) Assuming homogeneous capacity is used for the caches. However, our work can immediately extend to the network topologies where caches have different sizes (i.e., for the case where cache  $m \in [1 : N]$  has capacity  $K_m$ ).
  - c) Design a coordinated cache placement by partitioning video, CPVP, and assuming that all three parts of the video are available in all these three routers at the same time, our solution module will perform replacement based on Least Recently Used (LRU) mechanism. Saxena et al. [33, 163] emphasised that LRU holds high performance and has the potential of the increasing cache hit probability through the storage of the most recent data for extra time.

Our placement design calls the default replacement mechanism; whenever the data chunk arrives at the cache node, the conditions are satisfied ( $X > 3 \& R < 4$ ) and the

cache space is full. The study of Melchizedek et al. [312] stressed that LRU and Least Frequently Used (LFU) are the main cache replacement mechanisms, in addition to using just one parameter to make the substitution definitive. We will incorporate the LRU replacement mechanism into our design for the reasons mentioned previously. On the other hand, if LFU is used in our model, then CS occupation occurs with large-scale data due to these mechanism challenges, particularly the video, which will result in more time spent using LFU than LRU. Future networks are planning to integrate the ubiquitous caching program. Our new placement mechanism contains three schemes to support VoD networks and reduce network performance delays. The CPVP algorithm is known as 5.1:

---

**Algorithm 5.1** Coordination Caching Placement by Video Partitioning Mechanism (CPVP)

---

```

//Getting the initial values
Ri=get_Sequence_of_Router (CPIT);
X=get_total_hops_to_Source(CDC);
N=get_total_chunks_no_in_file V;
Ri=get_Sequence_of_Router (CPIT);
ID=get_content_id (CDC);
Chunk_SN=get_chunk_sn (CDC);
Data_chunk = get_Data_chunk_of_fileV;
On receiving CooperativeInterestChunk requesting i – th chunk of file V
if VAS() then // if the video aggregate is satisfied
    if VCDS() then call VPS() //if deciding to cache then partitioning the video
    ForwardCooperativeDataChunktoward consumer;//forward the data chunks to
        consumer
else// The video is still not aggregate then continue try to find the interested chunk
    Forward CooperativeInterestChunk toward source.

```

---

The CPVP mechanism does not require any centralized network management or additional high-burdened control overhead. It simply operates in every NDN's router in a distributed manner. CPVP conceptual model is shown in Figure 3.2. It shows how the VAS algorithm 5.2, the VCDS Algorithm 5.3, and the VPS Algorithm 5.4 are all linked in the flow chart. To achieve these goals, we update the basic tables of NDN by incorporating new fields, as detailed in the upcoming Subsection 5.4.1. The

general concept of how our design works is based on many steps that are clarified in this section. However, Section 5.4.2 goes into more detail about how to combine the video data chunks from the three routers. Subsection 5.4.3 explains how the video caching decision mechanism works, and Subsection 5.4.4 talks about how to partition videos into three parts sequentially.

#### **5.4.1 Create Cooperative Tables**

In NDN, especially during VoD applications, cooperative cache placement involves distributing video chunks across different caches in a network to improve efficiency and reduce latency. In this context, cooperative tables serve as shared data structures or information repositories, assisting multiple cache nodes in coordinating the caching and retrieval of video content. These tables hold crucial data regarding the caching of specific video chunks at different nodes, thereby guiding caching decisions to enhance performance.

Cooperative tables are pivotal in enhancing NDN cooperative caching because they facilitate effective coordination, sharing, and management of cached data among multiple nodes in the network. By ensuring efficient storage, retrieval, and management of content, these tables enhance caching performance. Cooperative tables are crucial to improved cache coordination due to their impact on reduced redundancy and increased efficiency in storing and retrieving content, leading to better use of available cache space. They are important for enhanced load balancing that leads to more reliable performance, particularly in high-demand scenarios, such as streaming services or VoD. Utilising these tables reduces unnecessary cache evictions and ensures the availability of the most valuable content, thereby improving cache utilization.

Cooperative tables facilitate cooperative caching, which lowers network latency by retrieving content from the closest cache. NDN nodes can improve content delivery

times and decrease overall user delay by using these tables to track the location and availability of cached data rather than fetching it from far-off servers. Cooperative tables help NDN networks scale by distributing cache management across multiple nodes. Instead of relying on a central server or cache, nodes can autonomously manage their caches and share information through cooperative tables, which allows the network to grow and accommodate more users without suffering from performance degradation. Thus, cooperative tables enable scalable caching solutions to handle increasing content demand without overwhelming individual cache nodes or servers. Cooperative tables are fundamental to enhancing NDN cooperative caching by providing mechanisms for cache coordination. By leveraging these tables, NDN networks can achieve greater efficiency, scalability, and user satisfaction, especially in high-demand environments like video streaming or VoD systems. Incorporating cooperative tables in the NDN framework addresses several significant challenges and limitations, making caching and content retrieval more efficient and scalable.

While cooperative tables greatly improve the performance and coordination in NDN caching systems, they also introduce challenges related to complexity, scalability, security, and privacy. Addressing these challenges requires careful design to balance the benefits of cooperation with the overhead and potential vulnerabilities that cooperative tables can introduce. We try to solve these challenges include:

### **1. Increased Complexity and Overhead**

Managing and maintaining cooperative tables across multiple nodes increases the system's complexity. Each node must share, synchronize, and update information regarding cached content, which requires frequent communication and resource sharing. This creates overhead in terms of bandwidth and computational power.

To address this issue, we only update our tables along the data's path. We base

the length path on the location of the data. In our design, the worst-case scenario involves finding the data on the server. This scenario occurs once for each unique request. The other scenario involves locating data on the intermediate node before it reaches the server. In the second scenario, our design employs a subset of three routers, so the consumer will find the data primarily in the first three routers. Therefore, data sharing will only occur on the subset of routers, not on the larger network. Our design adopts a simple condition to minimise sharing and complexity costs. Therefore, our design does not rely on popularity, eliminating the need to recalculate popularity for each request. In order to reduce bandwidth usage, our design employs chunking rather than segmentation, and in the second scenario, we only transmit the third part of the video file, rather than the entire file.

## 2. Scalability Issues

In highly dynamic networks with many nodes and content requests, ensuring that coordinated tables remain synchronized and up-to-date becomes a challenge. This can result in caching decisions using outdated or inaccurate information, leading to cache misses or inefficient data retrieval.

To solve this problem, our design does not share data across all nodes in a large network, but rather across a small number of routers. Our design boosts the hit ratio as demand rises by keeping data near the consumer and storing only one-third of the video in each of the three routers. These three routers are meeting the demand.

## 3. Inconsistent Cache States

Nodes in a cooperative caching system rely on the accuracy of shared information in the cooperative tables. Inconsistencies in cache states can arise if nodes fail to communicate effectively or update this information in real-time. Inconsistent cache states can lead to incorrect forwarding decisions, causing increased retrieval times and cache misses. For instance, a node may forward

a request unnecessarily, leading to increased latency, if it perceives another node to have cached a piece of content that is no longer valid.

To address this issue and prevent the loss of one-third of the video file, our design employs a caching strategy for each of the three parts at the same time, leveraging the LRU replacement mechanism to ensure data removal based on timing. This results in the near-simultaneous eviction of all three parts. In addition, the researcher highlighted that the default LRU is better than the other LFU replacement mechanisms in terms of latency.

#### **4. Security and Trust Issues**

Cooperative tables require nodes to share information about their cached content. In an open or distributed environment, malicious nodes could manipulate or provide false cache information, leading to inefficiencies or security vulnerabilities. Untrustworthy nodes can deliberately misreport cache states or redirect content requests to introduce delays or disrupt network operations. Ensuring data integrity and secure communication between nodes while sharing cooperative table information can be challenging, especially in untrusted or decentralized networks.

We can mitigate the challenge by sharing cooperative table information with a subset of routers rather than with all routers in the network. On the other hand, our design adopts NDN, which is a more secure network than the others.

#### **5. Resource Constraints**

Not all nodes in the network may have sufficient computational resources or bandwidth to maintain and update cooperative tables continuously. In resource-constrained environments, nodes may struggle to handle the overhead of maintaining cooperative tables, which could lead to slower cache decisions and suboptimal content placement. Limited resource nodes may also face delays in updating their cooperative tables, resulting in the caching of outdated information. Resource-constrained environments are characterized by limited

processing power, storage capacity, and network bandwidth, which poses a challenge.

Because of limited processing power, devices with constrained CPUs cannot handle complex computations, which require lightweight algorithms and optimized code. Thus, our algorithm is less complex than the other cooperated algorithms. With the limited storage capacity of routers, our design often requires effective data management and caching, which involves partitioning the cache video into three distinct blocks across three different routers. This approach enhances memory utilization and boosts diversity. The first point explains how to reduce network bandwidth.

## **6. Latency in Table Updates**

It can be challenging to ensure real-time updates to cooperative tables in fast-changing networks that frequently request or cache content. Sending requests to nodes that no longer have the cached content due to delays in updating cooperative tables could increase retrieval latency. Additionally, content placement decisions based on outdated information may lead to unnecessary cache evictions or inefficient caching.

Therefore, our design simplifies multi-request updating by eliminating the need for additional processes and complexity to update the cooperative tables, thereby avoiding popularity-based caching. Furthermore, we cache data close to the user, thereby reducing latency.

## **7. Privacy Concerns**

Cooperative tables require nodes to share information about the content they cache and serve. In networks where content requests are sensitive or the identity of the requesting node requires protection, this could raise privacy concerns. In scenarios involving sensitive data, the exposure of cache and content request information through cooperative tables could compromise user privacy. Managing cooperative tables securely while maintaining privacy poses a

significant challenge, especially in networks that serve a diverse range of content types. Therefore, our design serves VoD only rather than multiple types of content.

In our cooperated caching CPVP, the mechanism requires updating the basic tables as shown in Figures 5.5, 5.6, and 5.7. The initial data of all these records is equal to "0".

- **Cooperative Interest Chunk Table (CIC):** Our coordinated mechanism uses it as a fundamental component of the NDN structure. This study incorporates various variables, including the router ( $R$ ) and block ( $ID$ ) sequences. The router sequence is crucial in determining which router to cache. In VAS, the path increments  $R$  in order to locate data. Then, it was used in the VCDS condition to decide whether to cache or not. Additionally, it was used to determine which data chunk should be cached and which router should host it. Moreover,  $ID$  equals 0 if the content store had the entire video.  $ID$  can be set to 1, 2, or 3 to indicate the order of the blocks. Subsection 5.4.4 will provide a detailed explanation of how the VPS partitions each video into three blocks. Moreover, the CIC is utilised to transmit the relevant video information from one router to another, following a specific path until the video file is found.
- **Cooperative Data Chunk Table (CDC):** Our coordinated mechanism uses it as a fundamental component of the NDN structure. This study incorporates various variables, including the hop distance ( $X$ ), which is equal to the last value of  $R$  in VAS. Additionally, the VCDS condition uses the hop distance ( $X$ ) to decide which data chunk to cache and which router to use. The consumer receives the video file, its information, and the value of  $X$  through using CDC.
- **Cooperative Content Store Table (CCS):** We use it in place of CS, adding three fields:  $X$ ,  $ID$ , and  $N$ . The value of  $X$  represents the hop distance required

to retrieve the original data. This router caches a sequence of blocks known as *ID*. *N* represents the total number of chunks in this video file. When the VAS finds data with an *ID* of 0, it indicates that the data encompasses the entire video, and the VAS is satisfied. If the *ID* is 1, 2, or 3, then another request should be sent until the entire file, which contains *N* chunks, is aggregated.

- **Cooperative Pending Interest Table (CPIT):** Our coordinated mechanism uses this basic component instead of PIT in the NDN structure. This study added a field named *R*, denoting the router's sequence. We derive the value of this variable from *R* in the CIC table. VCDS uses it as a label to determine whether or not this router should cache the requested data. Therefore, we have added it to facilitate the ability to identify the three nearest routers to the user.

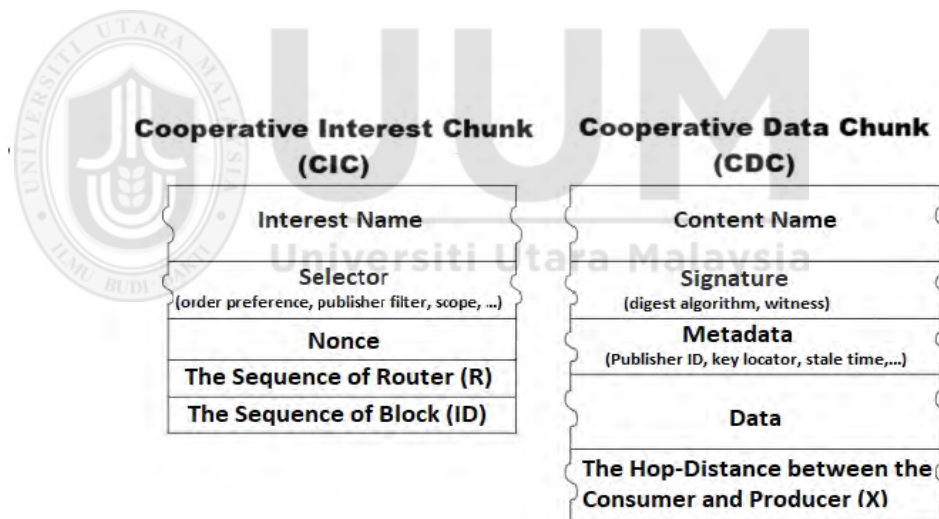


Figure 5.5. The Cooperative Chunks in NDN Tables: (CIC) & (CDC)

Cooperative Pending Interest Table (CPIT)		
Name	Requesting Face (s)	The Sequence of Router (R)
/parc.com/videos/WidgetA.mpg/v3/s1	CR A	1

Figure 5.6. The Cooperative Pending Interest Table (CPIT)

## Cooperative Content Store (CCS)

<b>Content Name</b>
<b>Signature</b> (digest algorithm, witness)
<b>Metadata</b> (Publisher ID, key locator, stale time,...)
<b>Data</b>
<b>The Hop-Distance between the Consumer and Producer (X)</b>
<b>The Sequence of Block (ID)</b>
<b>The Total Number of Chunks (N)</b>

Figure 5.7. The Cooperative Content Store Table (CCS)

### 5.4.2 Video Aggregation Scheme (VAS)

A quick scan within the network is very important when interest is served [313]. However, searching a long network will demand a fast approach to return the results of the subscribed object. The miss probability of each cache and the average number of response hops are used to evaluate the accuracy of a-NET [58, 314]. Thomas et al. [315] emphasised that the network offers better caching accuracy when it is getting accountable for most cache\_hits. Therefore, the proposed cache placement will be at its best accuracy because it has hit at each of the three nodes and these nodes are not having the whole video whereas caching the whole video will lead to high memory consumption. In order to show the accuracy of our proposed model, there are other challenges created, which include how to search and find the chunks of the requested video.

According to Aloulou et al. [65], the Basic NDN Forwarding mechanism [316] is the best forwarding mechanism because it offers a higher throughput with a lower latency time and uses a flooding routing mechanism [65]. Hence, the VAS uses the Basic NDN forward mechanism. Besides, with explicit cache coordination, content routers explicitly exchange their state and additional information with each other. Therefore, the proposed VAS will be merged with two other schemes, which are VCDS (in Subsection 5.4.3) and the VPS (in Subsection 5.4.4). This merging is achieved to design the CPVP mechanism as shown in Algorithm 5.1. The VAS is consisting of multi-steps explained below:

**First Step:** When the consumer achieves the request, this VAS mechanism will use the Basic NDN forward to reach the content producer. Designing a VAS serves two purposes: The first one is aggregating video; the second uses the field  $R$  as a label to find the proper router to cache video. This mechanism initials the  $R$  field to have the sequence of the router in the path between the consumer and the producer to assist in the proposed cache placement mechanism as explained above while putting this field in the  $CIC$  chunk to give the next router the value of  $R + 1$ . Furthermore, VAS gives a value to represent the sequence of blocks ( $ID$ ) in  $CIC$ . Note that the initial value of the  $ID$  will equal 0 when no block of data has been found. This field will then be used to gather another two blocks of requested data by knowing the  $ID$  of the detected partial data. Therefore, the  $CIC$  and  $CDC$  will be as in Figure 5.5 while the  $CPIT$  will be shown in Figure 5.6 after adding the new fields to the original one as in [293].

**The second step:** This represents the scanning phase that involves checking if the  $CCS$  contains the requested data or not. Then when confirmed to include the requested data, the  $CCS$  must check this data if it is the complete requested data or one-third of it. This content producer will either be a server that contains all the  $CDC$  of the request as shown in Figure 5.8 or is a router which contains one-third part of the request data.

Therefore, this step contains three cases which are:

A. There is a miss when the *CCS* does not contain the requested data. Then the *CPIT* of the same router will save the prefix name of the request, the requesting face, the sequence of the router ( $R$ ) and then forward the *CIC* with the value of  $R + 1$  until it finds the requested data chunks, while an *ID* is still set to zero ( $ID = 0$ ).

B. There is a hit, and the *CCS* contains the whole *CDC*, then the hop distance between the consumer and producer ( $X$ ) will be set at the same value of  $R$  as shown in Figure 5.8

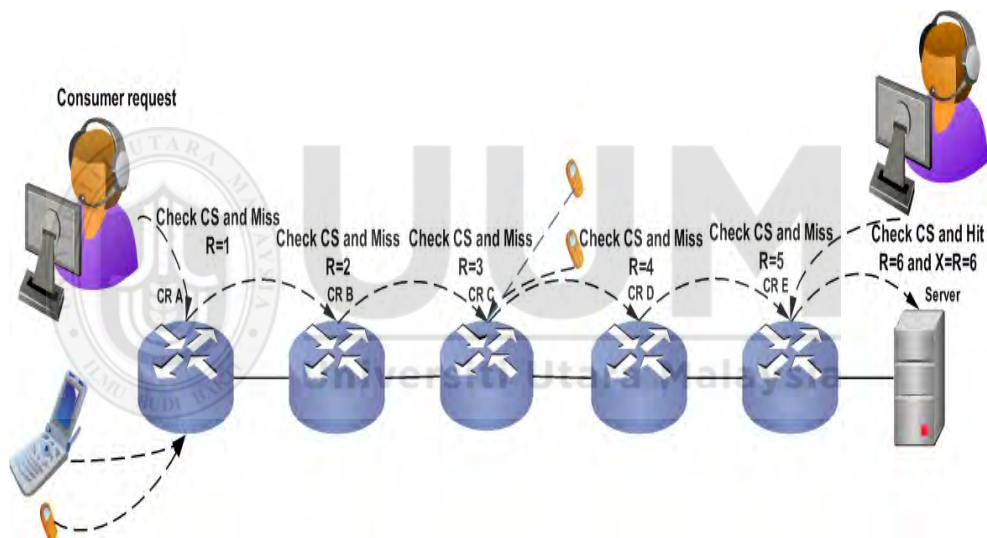


Figure 5.8. Request for Video and Search in CS Along the Path to the Server in NDN

C. There is a hit, but the *CCS* contains only one-third of the requested data (one block). This leads to bringing another two blocks of cached copies in the neighbourhood by forwarding the *CPIT* twice to the neighbourhood routers with the requested name and the sequence of the block (*ID*) to gather all data chunks of the same request together. Then a hit is registered, and  $X$  is set to the value of  $R$  ( $X = R$ ) when the requested data is gathered as shown the Figure 5.9.

**The third Step:** Forward the whole requested data with the value of  $X$  downstream in the reverse path to the consumer using and call our new *VCDS* mechanism which is the Video Caching Decision Scheme.

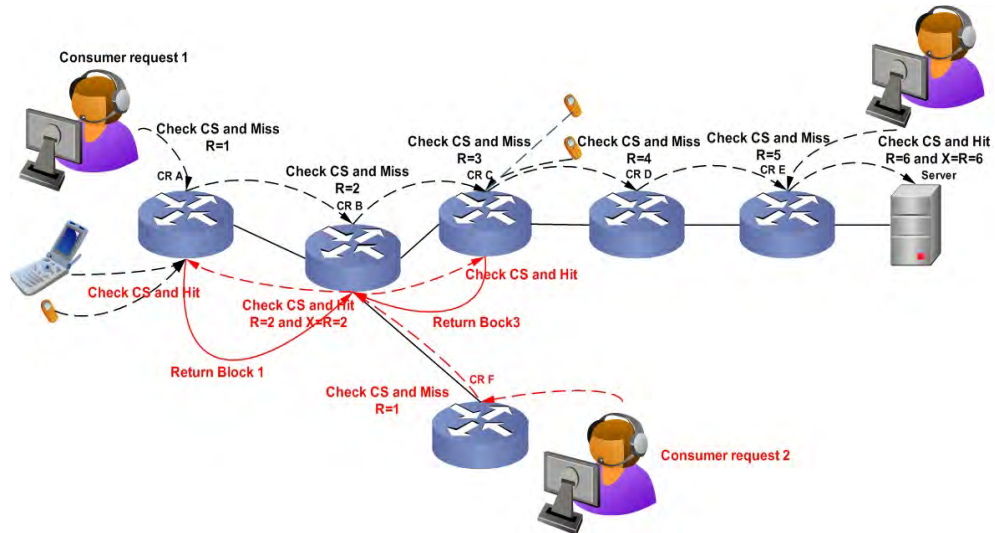


Figure 5.9. Gathered the Partial Data Chunks

Furthermore, when individual routers receive an *CIC*, it initially searches their *CCS*. If a hit exists, the *CDC* will be returned, or otherwise verified in the *FIB*. If a temporary path is present, the *CIC* will go through the path that was already discovered to the off-path data copies or looks for a match in *CCS* if otherwise. If it exists, it starts a new scanning phase and then, the *CIC* follows the associated interface until it finds the cached data. In the end, the scheme converges into a flooding-based searching scheme which introduces a vast amount of overhead. All the above steps are summarized in Algorithm 5.2.

---

**Algorithm 5.2** The Video Aggregation Scheme (VAS)

---

```
//Getting the initial values
Chunk_SN = get_chunk_sn (CDC);
Data_chunk = get_Data_chunk_of_fileV;
Ri= get_Sequence_of_Router (CPIT); X= get_total_hops_to_Source(CDC);
N= get_total_chunks_no_in_file V; R = 0; X = 0; ID = 0;

On receiving CIC requesting i – th chunk of file K

Repeat if CDC of K isn't in CCS then//Data chunk is not in CCS
    R = Ri + 1;
    save(prefix – name, requesting – face, R) in CPIT;
    Update R in CIC;
    call Basic NDN Forwarding CIC toward the server
until (CDC of K is in CCS)
if (CDC of K is in CCS) then
    ID= get_Block's_no(CCS); N= get_total_chunks_no_in_file V
    if ID = 0 & X = Ri then //Find the content in the server
        for j = 1 to N
            save X in CDC; call Basic NDN to forward CDCj to the consumer
        end_for
    if ID = 1 then
        for j = 1 to Ri * N/3
            save X in CDCj; call Basic NDN to forward CDCj to the consumer
        end_for
        send CIC(content – name, ID = 2); send CIC(content – name, ID = 3);
    if ID = 2 then
        for j = [((Ri – 1) * N/3) + 1] to Ri * N/3
            save X in CDCj; call Basic NDN to forward CDCj to the consumer
        end_for
        send CIC(content – name, ID = 1); send CIC(content – name, ID = 3);
    else //ID = 3
        for j = ((Ri – 1) * N/3) + 1 to N
            save X in CDCj; call Basic NDN to forward CDCj to the consumer
        end_for
        send CIC(content – name, ID = 1); send CIC(content – name, ID = 2);
    end_if
end_if
Delete Interest record from CPIT;
```

---

### 5.4.3 Video Caching Decision Scheme (VCDS)

There are multi conditions to cache the cooperative data chunk:

1. If the name in CDC is equal to the name in CPIT, this condition to check if this

router is requested this name of video. If the answer is “Yes” then check the next step, else Discard.

2. If the total chunk number ( $N$ ) of file is large than 3, and the total hop distance ( $X$ ) for this request is more than 3. Also, This condition follows because the data chunk proposed is to be divided to three nearest routers. Therefore, in this way, the data chunks are already near to the consumer. When  $X$  is equal to 3, the time will be wasted by trying to cache the nearest data. Whereas if  $X$  is less than three, it cannot find three routers to cache three blocks of chunks ( $Ch$ ) in the path to the consumer which leads to loss chunks from the requested video.
3. This condition involves checking each  $CPIT$  for every router in the reverse path if  $R < 4$ . If  $R < 4$  is satisfied, this means that the distance between the consumer and this router which is in the reverse path is three or less than three. Therefore, the video's chunks can be divided by the nearest three routers. While if  $R > 3$ , then this router in the reverse path is far from the consumer leading to waste time by achieving the partitioning and caching in addition to increasing delay, and experiencing multi replacement in  $CCS$  due to the large size of the video. Furthermore, the hop distance will increase if the cache is far away from the consumer when the next interest happens to the same request.
4. If  $condition1 \ \& \ condition2 \ \& \ condition3 = True$ , so decide to cache by call  $VPS$ , else forward  $CDC$  to the consumer.

If decide to cache then call  $VPS$ , else discard the data chunk, as shown in Algorithm 5.3.

---

**Algorithm 5.3** Video Cache Decision Scheme (VCDS)

---

```
//Getting the initial values
Data_chunk = get_Data_chunk_of_fileV;
Ri = get_Sequence_of_Router (CPIT);
X = get_total_hops_to_Source(CDC);
N = get_total_chunks_no_in_file V;
On receive CooperativeDataChunk to the router
name1 = get_name (CDC)
if name1 in (CPIT) then
    if  $N > 3 \ \& \ X > 3 \ \& \ R_i < 4$  then
        call Video Partitioning scheme (VPS)
    else
        forward CooperativeDataChunk to the consumer
    end_if
else
    Discard
end_if
forward CooperativeDataChunk to the consumer
```

---

#### 5.4.4 Video Partitioning Scheme (VPS)

The router receives the majority of file requests in most cache placement mechanisms, including LCD, LCE, and Prob. Because the most popular files are cached wholly at the router, the logic behind splitting video files is to offer the other routers a chance to get requests for a portion of the files.

This concept might not seem practical at first because the file will be cached across multiple routers and downloading the complete file might take longer. Nevertheless, there are two major reasons in favor of this choice:

1. Caching a part of a file allows additional files to become closer to the clients. In other words, an edge router can find a subset (a few parts) of each file if there are numerous files in the location.
2. A video file is typically a huge file worth caching. When downloading a video clip, the time it takes to download the first part, which is closest to the consumer,

can overlap with the time it takes to get the remaining parts, which are cached inside the network.

To fully utilise the idea, the *VAS* is described in Section 5.4.2, and *VCDS* is presented in Section 5.4.3. Then, the results are given from experiments in Section 3.3

Clients in NDN demand content by sending an Interest that explicitly deals with a Data chunk of a content file. The content in this thesis treats video only, which is considered as files.

#### 5.4.4.1 Description of VPS

Each NDN router has a content store (CS) for caching Data chunks. (we assume three routers participate in caching). By default, NDN caches a chunk at every router that the chunk visits [117]. This redundancy wastes caching capacity, especially when CS size is already small compared to the number of files. Moreover, hits at edge routers filter out their Interests, so copies in the core get few hits [317], which leads to wasting capacity.

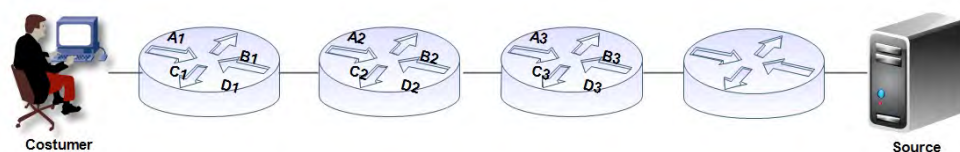


Figure 5.10. Partition File on the Path

Selectively caching popular content at the periphery and unpopular content at the core could be a way to improve performance. It is impractical to do this using the video file popularity because it involves ranking a lot of video files instantly as they are added and their popularity changes. Additionally, because the popular files are huge, they push out the nearby ones. As an alternative way, the VPS caches a file over several routers along the identified path between the client and the source, with the head closer to the

customer and the tail closer to the source ( see Figure 5.10). There are three reasons for doing so:

1. Users frequently cancel downloads, according to earlier traffic studies [318]. Therefore, chunks on the top of a file are more sought-after than those at the bottom. Thus, VPS is consistent with finding that popular content eventually caches at the edge of a network [317, 319].
2. The close cache space to the clients is precious. If the head is cached only at the edge, this significant resource can be shared by additional files.
3. If the content is a video, user-perceived latency is based on how the application soon can get enough of the video file to get startup. Delays in recovering the rest of the file can be hidden during taking time for the application to play back the head; it is a similar concept in the solid-state disks to the file layout [320].

Although the customer propensity issue to cancel is one of primary drivers for partitioning files on the route, thus, it into account in our trials because of unable to identify any logical distribution to capture customers' propensity to cancel downloads. However, later experiments will show that the first routers closer to the customer are more valuable than the other routers. Terminating a download prevents the tail of the stream from being cached on the routers closer to the source, which is valuable to clients nearby. So, from this perspective, VPS's performance in the real world may even be superior to what we demonstrate through the experiences.

The VPS divides a video file into three blocks. Each block (multi-chunks) is cached at a different neighbor router. The notation utilised is shown in Table 5.2.

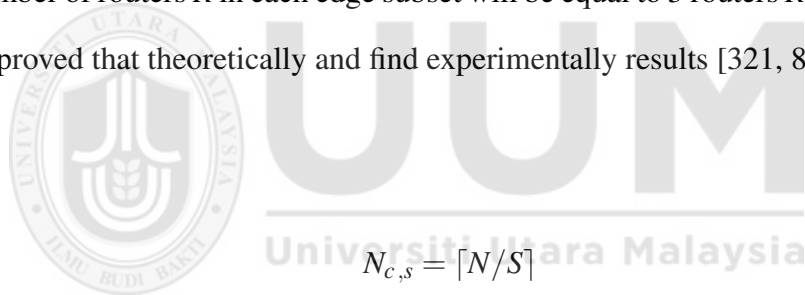
How should be partitioned a video file? On the route from the customer to the source,

**Table 5.2**

Notation for Video Partitioning Scheme (VPS)

Parameter	Notation
$S$	Number of segments in a file $F$ .
$X$	Hop count to reach the source.
$R_i$	Hop count refer to the reached router before a source.
$N$	Number of chunks $c$ in a file $F$ .
$N_{c,s}$	Number of chunks $c$ in a segment $s$ .
$N_s$	Number of segments $s$ cached at router for a file $F$ .

we could employ a hop count  $X$  by dividing a file into  $R_i - 1$  segments to cache segment  $s_i$  th at router  $R_i$  th on the path from customer to source. However, if two customers with various hops counts request the same file, this leads to getting various segment sizes, and don't share segments. As an alternative, VPS ensures that each file  $F$  contains  $S$  segments, whereas the size of each segment is in Equation 5.18 and fixes the number of routers  $R$  in each edge subset will be equal to 3 routers  $R = 3$  as Dehghan et. all proved that theoretically and find experimentally results [321, 82, 301].



$$N_{c,s} = \lceil N/S \rceil \quad (5.18)$$

VPS uses the additional parameter  $X$ , which is the maximum hop count between the customer and the source (the server). Suggest an Interest  $T$  from customer  $C$  takes a path over the routers  $R_1, \dots, R_{X-1}, \dots$  to arrive at the source, wherever  $R_i$  takes  $I$  hops from the customer  $C$ . Suppose that  $N_s = \lceil S/3 \rceil$  then each router caches no more than  $N_s$  segments. Therefore;  $R_1$  caches 1st, 2nd, 3th, ...,  $N_s$  segment, while  $R_2$  caches the series of segments  $(N_s + 1, \dots, 2N_s)$ , etc. In other word,  $R_i$  contains series of segment starts from segment number  $((i - 1) * N_s) + 1$  until segment number  $(i * N_s)$ .

For example, suppose that the number of segments per file is  $S = 21$  and the number of total hop accounts is  $X = 22$  for a pair of requester-source. If file  $F$  contains 1024

chunks, then each segment contains about  $1024/21 \simeq 49$  chunks (except for the last segment); while the number of segments per router contains  $N_s = 21/3 \simeq 7$ , so  $R_1$  caches segments 1, 2, 3, 4, 5, 6 and 7, while  $R_2$  caches 8, 9, 10, 11, 12, 13, and 14, and so on. In another case, if the file is partitioning to all routers on the path to the source, then  $N_s = \lceil S/X - 1 \rceil = (21/21) = 1$  segment per each router to the source. This case may lead to losing data, and the last segment is far away from the customer, which needs a lot of additional time, which leads to increasing the delay time. By consuming VPS depends on number of chunk, therefore each three routers will contain block of chunk starts from the  $Chunk - SN$  as a result of Equation 5.19:

$$Chunk - SN_s = ((R_i - 1) * N/3) + 1 \quad (5.19)$$

While each three blocks will end with  $Chunk - SN$  as a result of Equation 5.20:

$$Chunk - SN_e = R_i * N/3 \quad (5.20)$$

Therefore, if combine these two equations Equation 5.19 and Equation 5.20 for three routers, will obtain Equation 5.21:

$$CacheChunk - SN = \begin{cases} Chunk - SN_s \leq Chunk - SN \leq Chunk - SN_e & \text{If } R_i = 1, \text{ or } R_i = 2 \\ Chunk - SN_s \leq Chunk - SN \leq N & \text{If } R_i = 3 \end{cases} \quad (5.21)$$

How can a router determine which segments to cache? In NDN, there isn't an addressing system for finding a router. However, as a chunk's path precisely reverses that of its Interest, we may utilise hop count to determine a router by doing the following:

An Interest  $T$  from customer  $C$  searches the CS in every router along its path to the source. During this search,  $T$  keeps track of the hop count on its path from  $C$  to the source.  $T$  may find its Data along the path, thus not reaching source  $Z$ . If  $T$  arrives at  $Z$ ,  $Z$  knows the hop count  $X$  from  $C$ . It then calculates  $N_s$ .  $Z$  then puts  $X$  and  $T$  in each chunk's header. As the chunk passes through a router on the reverse path to  $C$ ,  $X$  is decremented. This value becomes equal to  $i$  at router  $R_i$ , so if  $R_i \leq N_s$ ,  $R_i$  knows it has to cache chunks in determined  $N_s$  as explained above; other routers do not cache these chunks as it passes through (this is also valid if the Data source is an intermediate router).

When a cache is full, VPS uses LRU (Least Recently Used) mechanism to make replacements because this mechanism is close to optimal in experience [322] In addition, it depends on the time to replace the content, while we assumed that caching each part from the same data is achieved at the same time on different routers.

#### **5.4.4.2 VPS Mechanism & Flowchart**

This mechanism is proposed to cache the chunks on the edge-side cache nodes for a reduction in user-perceived delay, and a cache pipeline is progressively built on the path between the content origin server and the user. This placement mechanism will not base popularity because the content popularity is recorded and ranked with respect to the number of historical requests. Thus, caching based popularity leads to occupying memory. However, this approach for chunk-level popularity has its shortcomings, in particular when considering video applications. In an extreme scenario, the same video

file is requested by all users at the same time. In this case, the formerly requested chunks having a higher number of previous requests are categorized as popular and selected to be cached for a long time which is an occupation problem that leads to more delay in the network.

Moreover, the inter-chunks relationship is not appropriately analyzed in past studies. For this reason, the behavior of the cache focuses specifically on a single data. The domination of the traffic in NDN by video applications makes it necessary to investigate the relationship between video chunks. For instance, if a video file is divided into chunks having the same name prefix and growing segment numbers (affixed to the name of the content). That is, a video file  $F_i$  is divided into  $K$  ( $K \in N$ ) chunks with a name prefix  $F_i$  and sequenced indexes, i.e.,  $\{Ch_1^{f_i}, Ch_2^{f_i}, \dots, Ch_j^{f_i}, \dots, 0 - 1hC_k^{f_i}\}$ , where  $Ch_j^{f_i}$  is the  $j$ -th chunk  $j$  of file  $F_i$ . It is important to maintain the proposed cache Placement mechanism based on path explicit coordination caching scheme, and the following steps in Figure 5.8 below show how this is achieved.

**1. Get the hop-distance (X) from consumer to the producer and also find the distance from each router to the consumer (R):**

Finding the interested file involves searching the entire CS on its path by using **Video Aggregation Scheme (VAS)**. The returned data packet contains the hop distance ( $X$ ) to the router where the data chunks were found. A router will decide to cache the data chunks of the interested file in the CS table only if its hop distance ( $X$ ) from the consumer to the producer is greater than three. In this way, since FIB is slower than PIT [82], then it is suggested to add more information ( $R$ ) to the interest request in the PIT which represent the number of hops between the router and consumer who requests the interest. This is achieved by increasing  $R$  by one with each hop and saving

its field at the PIT table of the same request until the requested file in the router is found (in other words, until the hit happens). An interest carries the hop count information, which is set at a default value of zero at the content requestor and increases by one any time an NDN router relays the interest. In this way, when the router has a hit to the same request,  $R$  will be equal to the whole hop distance between the consumer and the producer ( $X$ ). Also, it will be equivalent to the distance between the client and the arrived router, as shown in the following Figure 5.8.

In the above Figure, we can see that the consumer requests a video, then the search in the NDN is started by checking each CS in the path if it has the video or not. If the CS does not have the requested video, then the forwarding is continued with finding the value of  $R$  for each router until the hit happens. If the hit is achieved, then the value of  $X$  will be found, which represents the total hop-distance. That is, the value of  $R$  will be saved in the PIT, with the prefix name of the interested video and requesting face (such as CR A) inside the same router along the path. The name of the received data, the data, and the value of  $X$  will be saved in the data chunks in addition to the metadata because the interests and data chunks carry metadata information about the content file (name, size, etc.) as shown in Figure 5.5.

**2. Downstream the video with its  $X$  value and its corresponding information in reverse, back to the consumer:**

To name individual DASH-like video segments with multiple representations, [142] proposed a mechanism for mapping them to the naming system in NDN. Naming a DASH segment at a specific representation follows the definition in [61, 102] given as:

[Video content name]. [Representation ID]. [Segment ID]

With the possibility to leverage existing HTTP infrastructures at scale, MPEG-Dynamic Adaptive Streaming (MPEG-DASH) offers an open standard which comprises of the specification of an XML-based manifest known as a Media Presentation Description (MPD). As its name implies, the MPD gives a description of the relationship existing between the segment locations with the use of HTTP URLs and its corresponding features (such as bitrate, resolution, codec), and timeline [142]. Whereas, Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable [323]. In other words, in MPEG-DASH, these features are described by MPD, which is an XML document. With the records contained within the MPD, the DASH user can begin the streaming session and acclimatize to bandwidth fluctuations. The content's existence on the server is in two forms. The first is the MPD, describing a manifest of the existing content, its alternatives, their corresponding URL addresses, and other features, while the second form is the segments, which comprise of the main multimedia bit streamed in chunks form, be it as a single file or multiple files [324]. MPEG-DASH is the outcome of different proprietary forms developed by the industry, and research places interest on its optimization with respect to efficient delivery and QoE.

ICN is in its early stage and mostly pursued currently by research institutions. However, this field shows signs of being very promising in the area of efficient multimedia content delivery if widely implemented in the future Internet. DASH and its founding technology are essentially uncertain in terms of the delivery infrastructure, which currently is confined to HTTP. Although it scales nicely, however, it can be extended easily to a different method of transport such as ICN. DASH was proposed over CCN/NDN architecture alongside open source tools and datasets with the use of the already existing open source CCNx implementation as a basis [142]. Using the MPD, a DASH client can download the most appropriate segment satisfying the

users' context. There are already triple of works on adaptive video stream over CCN, namely [325], and [326], then [327]. Network coding in CCN has been shown to reduce network bandwidth and delay where a mobile device can associate. Therefore, all these works improve the QoE as well as mobility of the video streaming user with multiple radio access networks, potentially at the same time [328].

Samain et al. emphasised that many over-the-top mechanisms emerged recently for efficient video delivery over the Internet. These mechanisms are both proprietary such as Microsoft HSS and Apple HLS, as well as standard-based as in the case of Dynamic Adaptive Streaming over HTTP (DASH). Likewise, to bring about a reduction in the pressure that the video currently represents on the network infrastructure, authors also observed that there is an increasing offer of network frameworks, such as Server and Network Assisted DASH or Information-Centric Networks (ICN), together with network functions, like in-network caching and multi-path forwarding, which aim to assist in achieving effective video delivery [329].

**3. Cache the requested video partially in three nearest routers to the consumer**

**VCDS returns true.**

**4. This is new cached placement is achieved by divided the video to three blocks.**

Each block must cache in one of these three routers nearest to the consumer. In addition to cache the value of  $X$ , the total number of chunks in this video ( $N$ ), the partial number of chunks ( $C$ ) in this router ( $R$ ) sequentially, and  $ID$  number are also put in consideration. Where the  $ID$  number will be 1, 2, and 3 to 1st block, 2nd block, and 3rd block respectively. The benefits of  $ID$  will appear in the proposed Search of Chunks mechanism to know which block is found and which other two blocks need to continue a search. This partitioning of video file into three blocks will get two

situations based on the number of data chunks' video as explained below in these two examples:

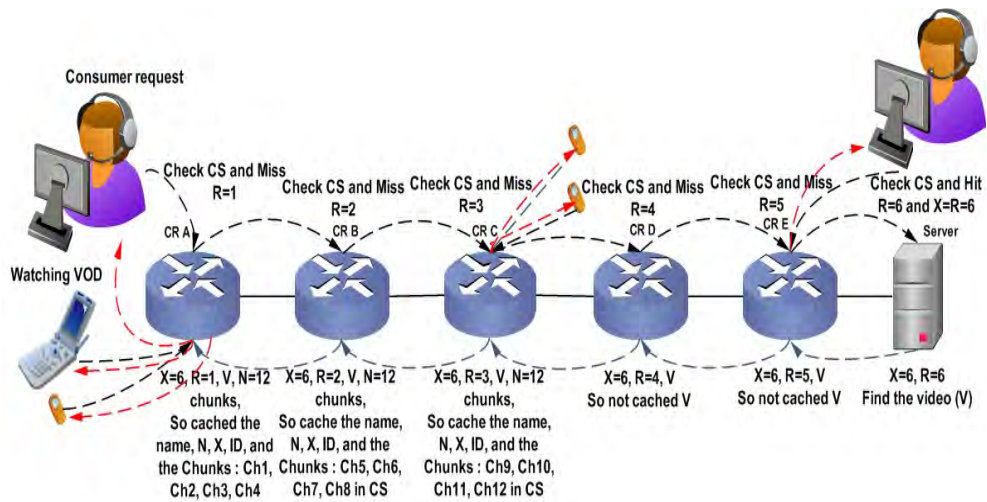


Figure 5.11. Case1 When the Video Can Be Divided to Three Blocks

**A. The first case:** The three blocks of three nearest routers contain different number of chunks to the same video. This happens when the total number of chunks cannot be divided by three without remainder. For example, assuming that the chunk is represented by  $Ch$  and the whole requested video ( $V$ ) contains 11 chunks ( $N = 11$ ), then 6 the nearest three routers (i.e.,  $R_1$ ,  $R_2$ , and  $R_3$ ) to the consumers must have different number of chunks in each block. In this example, each block must have four chunks ( $C = 4$ ) for the same video file by caching  $Ch_1$ ,  $Ch_2$ ,  $Ch_3$ , and  $Ch_4$  in  $R_1$ , while  $Ch_5$ ,  $Ch_6$ ,  $Ch_7$ , and  $Ch_8$  will be cached in  $R_2$ , and  $Ch_9$ ,  $Ch_{10}$ , and  $Ch_{11}$  will be cached in  $R_3$ . In this case, the question arises about why there is caching of the least number of chunk in the third router from the consumer? The answer here is that most consumers do not finish watching a video, and also because the first chunks are most important than the least chunks of the same video. The second question is about the reason for caching the chunks of the requested video sequentially and not randomly? The answer is to avoid missing any chunk, to make the video have high quality, and to avoid spending the time to rearrange the chunks of the video when the same video

requested again. This is shown in Figure 5.11 above.

**B. The second case:** is the three blocks of three nearest routers contain different number of chunks to the same video. This happened when the total number of chunks cannot be divided by 3 without remainder. For example, let us consume that the chunk is represented by  $Ch$ , and the whole requested video ( $V$ ) contains 11 chunks ( $N = 11$ ), then the nearest three routers (i.e.  $R_1$ ,  $R_2$ , and  $R_3$ ) to the consumers must have different number of chunks in each block. In this example each block must have four chunks ( $C = 4$ ) for the same video file by caching  $Ch_1$ ,  $Ch_2$ ,  $Ch_3$ , and  $Ch_4$  in  $R_1$ , while  $Ch_5$ ,  $Ch_6$ ,  $Ch_7$ , and  $Ch_8$  will be caching in  $R_2$  in order to cache  $Ch_9$ ,  $Ch_{10}$ , and  $Ch_{11}$  in  $R_3$ . In this case, question will be arising about why caching the less number of chunk in the third router from the consumer? The answer here, is because almost consumers don't finish watching video, and the another reason is the first chunks have the major important than the latest chunks of the same video. Second question is about the reason of cached the chunks of the requested video sequential and not randomly, and so on? The answer is to avoid missing any chunk, to make the video has high quality, and to avoid spending time to rearrange the chunks of video when the same video requested again, as shown in Figure 5.12 below:

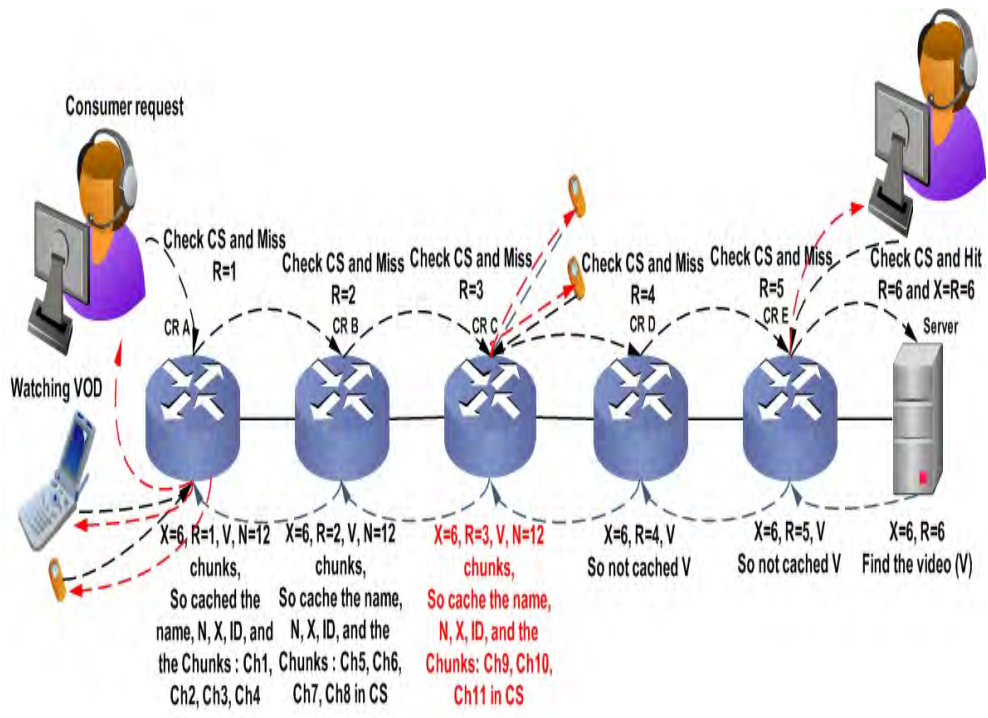


Figure 5.12. Case2 When the Data Chunk Cannot Be Divided to Three Block Without Remainder

and the following Algorithm 5.4 shows how this is achieved.

---

**Algorithm 5.4** Video Partitioning Scheme (VPS)

---

```
//Getting the initial values
Ri = get_Sequence_of_Router (CPIT);
ID = get_content_id (CDC);
Chunk_SN = get_chunk_sn (CDC);
N = get_total_chunks_no_in_file V;
Data_chunk = get_Data_chunk_of_fileV;
if Ri = 3 & Chunk - SNs ≤ Chunk - SN ≤ N then
    if CCS is not full then
        ID = Ri;
        cache DataChunk, ID in CCS;
    else
        ID = Ri;
        call replacement mechanism LRU to DataChunk, ID in CCS;
    endif
else if (Ri = 1, or Ri = 2) & Chunk - SNs ≤ Chunk - SN ≤ Chunk - SNe then
    if CCS is not full then
        ID = Ri;
        cache DataChunk, ID in CCS;
    else
        ID = Ri;
        call replacement mechanism LRU to DataChunk, ID in CCS;
    endif
endif
forward CooperativeDataChunk to the consumer
```

## 5.5 Verification and Validation of Coordination Caching Placement by Video Partitioning Mechanism (CPVP)

Designing the new mechanism, covered in Chapter 5 in great detail, involves stating the design objectives. Three separate supporting mechanisms make up the new design. Verification is carried out by assessing the outcome to see if it adheres to the original objective or formulation. Furthermore, the CPVP mechanism implementation is validated on ndnSIM2.0 and it is compared to three other placement mechanisms, which are LCE, LCD, and Prob placement mechanisms.

Utilising random video sequence kinds, the **validation** is carried out. As previously mentioned, several video segments are combined in the video test sequence. As was

mentioned in **Chapter 6**, the outcome of the brand-new mechanism has worked as expected. Thus, it may be said that the new mechanism has been proven to work.

As mentioned earlier in this chapter and Chapter Four, all simulations were carried out randomly from a long enough database to cover all video sizes. **Verification** of the new mechanism (CPVP) was done to ensure that:

1. The new mechanism has been correctly programmed.
2. The new mechanism has been correctly implemented.
3. The new mechanism has no flaws, errors, or omissions.
4. There have been no implementation errors, and the specification is comprehensive.

This validation or verification often belongs in a protocol model validation or verification. From this perspective, several conventional approaches have been used to model validation and verification. These included testing, thorough code reviews, and organised programming. For situations with known outcomes, one important validation may be evaluated by comparing the new design output (CPVP) with the anticipated outcomes and conducting a thorough trace of the event sequence throughout a model execution to ensure it follows the predicted route. In that regard, several mechanisms have been used to verify and validate the created protocol.

As described by Zakiuddin et al. [330], there are three kinds of automated formal verification mechanisms. They are as follows:

1. Model checking: Is the process of determining if a formal model system meets a specific property.

2. Theorem proving: As seen in this chapter, the theorem-proving methodology employs formulas, equations, and mathematical principles to demonstrate that a system is right.

3. Finally, equivalence testing explicitly determines if two models with various levels of abstraction are equal. This approach essentially compares two different systems and models. The new mechanism (CPVP) is compared with three other placement mechanisms (LCE, LCD, and Prob) in the next chapter.

According to Kristensen and Jensen [331], mechanism checking verifies every possible input to ensure that the system is accurate. However, the theorem-proving approach was used for the CPVP mechanism verification and validation. As a result, the mathematical equations discussed in **Chapter 5** are used to validate and verify the implementation of CPVP. This validation is achieved by using ndnSIM2.0, and a random video. This is achieved by using a Tree topology with 31 nodes, as shown in Figure 5.13. To prevent repeating the writing, **Chapter 6** will evaluate the CPVP placement method by adding more nodes and using different topologies.

## 5.6 Summary

This chapter deals with the second objective, which is designing a new caching placement mechanism in NDN during VoD. This new mechanism is called the Cooperative Caching Placement by Video Partitioning mechanism (CPVP). It contains three schemes to enhance the average hit ratio, the total reduction in the network footprint, server load reduction, and to reduce the average E2EDelay. We conduct a theoretical analysis of the content placement problem to determine that coordination caching outperforms non-coordination caching during VoD. These schemes are the Video Aggregation Scheme (VAS), Video Cache Decision Scheme (VCDS), and Video Partitioning Scheme (VPS). This mechanism proposed partitioning the video into three

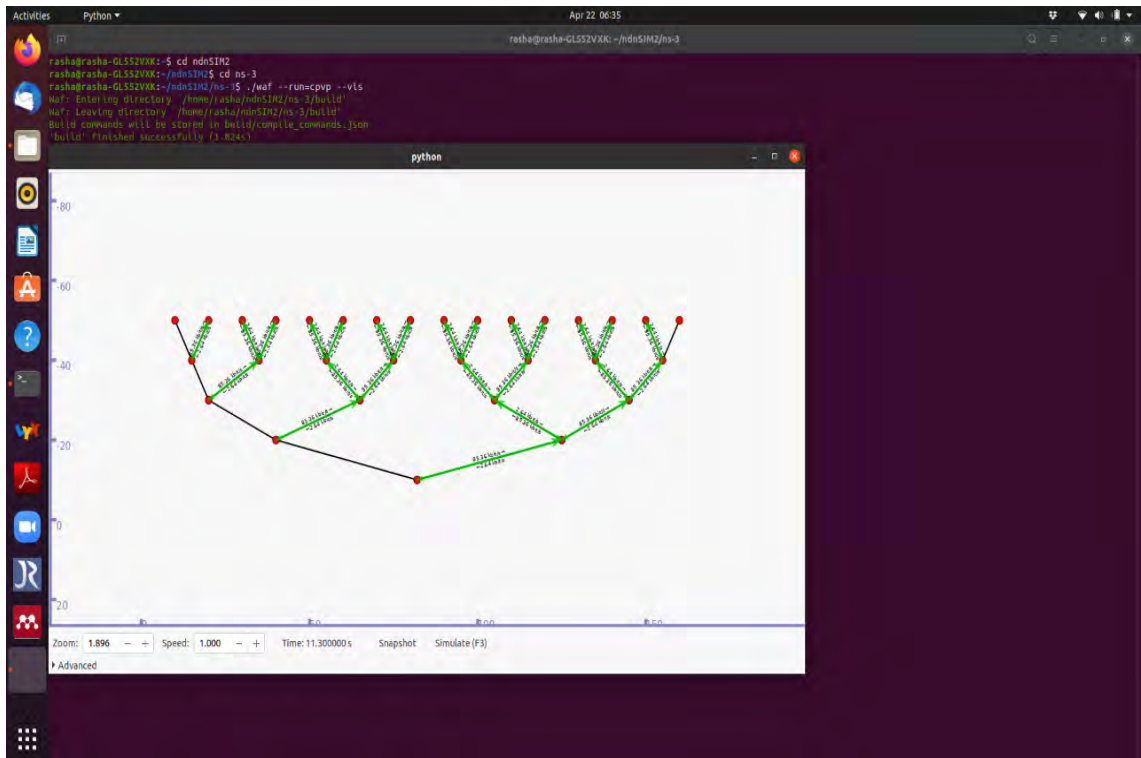


Figure 5.13. The CPVP Implementation in ndnSIM2.0

blocks and caching them in the three nearest routers to the consumer. Finally, the CPVP mechanism is verified, validated, and implemented at the end of this chapter, while it will be evaluated in the Sixth Chapter by comparing it with LCE, LCD, and Prob caching placement mechanisms to get the third objective.

## **CHAPTER SIX**

### **SIMULATION RESULTS AND EVALUATION**

Chapter Five presents the implementation of the Coordination Placement by Video Partitioning mechanism (CPVP). This chapter is to evaluate the CPVP mechanism during VoD in NDN. CPVP is an explicitly coordinated mechanism that includes three schemes: the Video Aggregation Scheme (VAS), the Video Caching Decision Scheme (VCDS), and the Video Partitioning Scheme (VPS) for VoD in NDN. This discussion of CPVP aims to discover if CPVP satisfies the functional requirements to enhance Quality of Experience (QoE). This chapter discusses the performance assessment of the CPVP mechanism and compares it with the LCE, LCD, and Prob caching mechanisms. Section 6.2 presents the evaluation and discussion of the CPVP mechanism, while Section 6.3 summarises the extent of CPVP's improvement. Section 6.4 provides a summary of this chapter..

#### **6.1 Simulation Environment**

Researchers often analyse their design methods for caching mechanisms by their simulation scenarios and select the configuration settings of their choice, making it impossible to compare the differences across caching mechanisms. Even though various simulation scenarios have been put forth for assessing NDN caching mechanisms, in this section, we focus on the same four primary simulation parameters as in [231]: network topology, catalogue size, cache size, and content popularity model.

The simulation results are elaborated in this section in line with performance metrics. Efficient evaluation of caching strategies in NDN requires simulation over diverse and representative network topologies, each capturing different structural properties (e.g., node centrality, path diversity, edge density) that significantly impact cache placement

and cooperation dynamics. Then, each performance metric will be discussed, as well as the enhancement percentage, if there is an enhancement for the new caching placement mechanism (CPVP).

### 6.1.1 Simulation Setup

To evaluate the new mechanism, CPVP, the performance metrics results are compared with these placement mechanisms: LCE, LCD, and Prob, to confirm that the simulation results of CPVP are valid. The reason for comparison with LCE, LCD, and Prob is that LCE is the default NDN caching mechanism; LCD is developed from LCE, while Prob is getting a good result in performance metrics in **Chapter Four** of this thesis. Additionally, the other reason is that each of these three mechanisms can be considered an example of one type of caching mentioned in **Chapter Two**. LCE mechanism is non-coordination on path caching as mentioned in Subsubsection 2.3.2.1 and explained in detail in Subsection 2.4.1. Whereas the LCD mechanism is an implicit coordination, as mentioned in Subsubsection 2.3.2.2 and explained in detail in Subsection 2.4.2. Prob is also in the same type of LCD, but it efficiently decreases redundancy, and big minimises the End-to-End Delay (E2EDelay) performance metrics. Therefore, our design represents the explicit cache coordination on the path. For more precise, see Figure 2.3.

The evaluation of our implementation by cache size from 100 content elements to 100,000 elements (1GB to 1TB sequentially), a catalogue size of  $10^6$  elements, while a Zipf popularity equal to  $\alpha = 0.65$  (see Table 3.2). In the literature,  $10^4$  contents is the widely utilise d catalogue size [260, 261, 269, 268], But because of the current traffic nature, it isn't easy to guess the Internet simply with  $10^4$  contents. As a result, in our studies, the catalogue size was set to 106 items, which is thought to be sufficient for assessing the performance of Internet traffic [233, 262]. Files are represented by the components and contents. As a result, they are both used equally in this thesis. In the

simulation outcomes, the median values of ten runs are evaluated for Average Cache Hit Ratio, Average E2EDelay, Server Load Reduction, and Reduction in Network Footprint. The basic NDN Forward mechanism is used according to considered as the best forwarding mechanism by the study of Aloulou N. et al [65]. Assuming that the CPVP is cached in the three blocks in the three neighbour routers at the same time, therefore, this study used Last Recently Used (LRU) as a caching replacement mechanism. However, each block is equal to 1/3 of a file . This study achieved simulation by using these topologies: Tree, Abilene, GEANT, and Deutsche Telekom Network Topologies in ndnSIM2.0. The VoD traffic is obtained randomly from the Scalable Video Coding-Dynamic Adaptive Streaming HTTP (SVC-DASH).

## **6.2 Simulation Results and Discussion**

The simulation results are elaborated in this section in line with performance metrics. Efficient evaluation of caching strategies in NDN requires simulation over diverse and representative network topologies, each capturing different structural properties (e.g., node centrality, path diversity, edge density) that significantly impact cache placement and cooperation dynamics [294]. Then, each performance metric will be discussed, and the percentage of enhancement, if there is an enhancement for the new caching placement mechanism (CPVP).

### **6.2.1 Average Cache Hit Ratio by Using Four Topologies**

The average cache hit ratio result is the most critical metric in the performance assessment of NDN. It is used to evaluate the CPVP mechanism by ndnSIM2.0 simulation and various cache sizes (1GB, 10GB, 100GB, and 1TB) in four topologies (Tree, Abilene, GEANT, and DTelekom) as shown in the Figures below. While the size for each file is taken randomly from the popular/established dataset which is the Scalable Video Coding-Dynamic Adaptive Streaming HTTP (SVC-DASH).

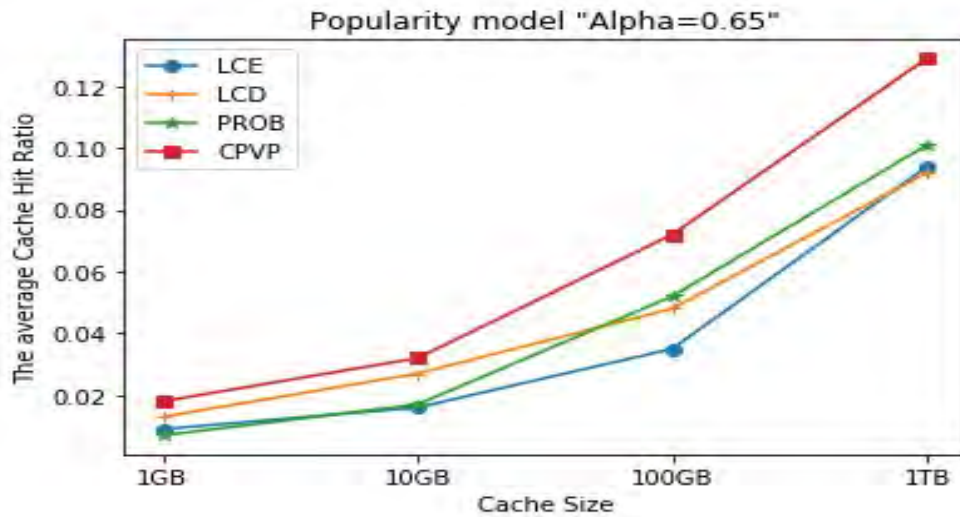


Figure 6.1. Average Cache Hit Ratio Using Tree Topology

Figure 6.1 represents the simulation results of the new mechanism, CPVP, and the other three placement mechanisms: LCE, LCD, and Prob. This results is achieved with using Tree topology. It is clear that the CPVP mechanism gets a better result than the other three placement mechanisms. Concluding that the average cache hit ratio is increased while the size of the cache is increased.

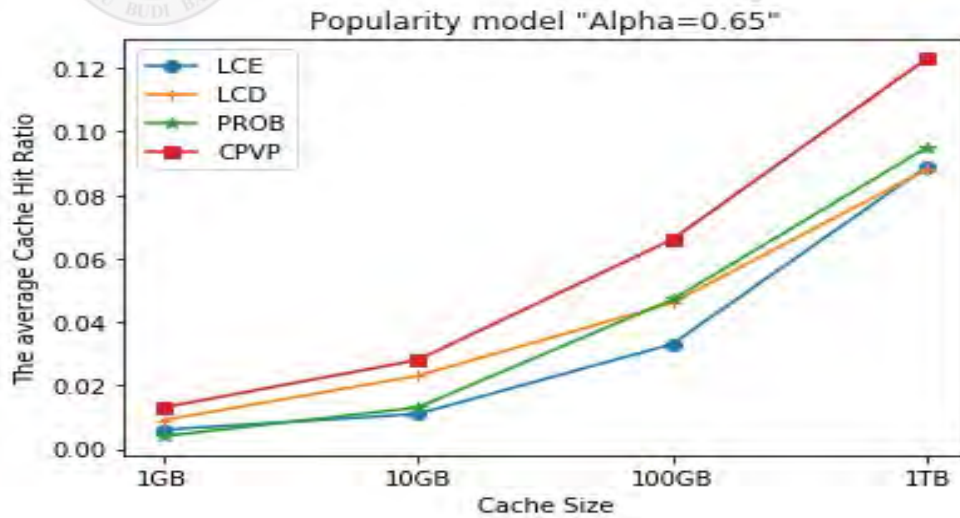


Figure 6.2. Average Cache Hit Ratio Using Abilene Topology

It is clear that in Figure 6.2, the average cache hit ratio of LCD is best than the other two mechanisms LCE, Prob in 1GB and 10GB due to its selective caching strategy. By

caching content closer to the requester and avoiding redundant caching, LCD achieves a higher cache hit ratio, especially in environments with limited cache space. LCE's redundant caching and Prob's randomness in caching decisions lead to less efficient cache use and lower hit ratios, whereas LCD effectively balances cache efficiency [332, 333]. While it is start to be closest to Prob in 100GB then closest to LCE in 1TB. But the CPVP mechanism is the best along with the increasing cache size and using Abilene topology. Whereas, it got 0.013 in 1GB and 0.123 in 1TB.

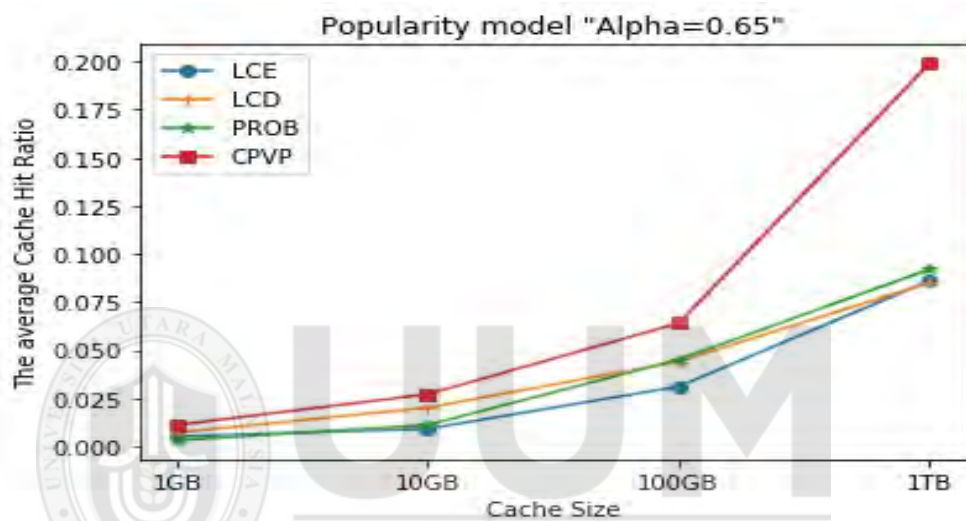


Figure 6.3. Average Cache Hit Ratio Using GEANT Topology

Figure 6.3 shows the results of an average cache hit ratio by using the GEANT topology to appear that the CPVP mechanism got the highest average hit ratio. The results of LCE is the worst results of the average cache hit ratio along various cache size because LCE causes multiple nodes to cache the same content and leading to redundant storage of popular chunks, and limited space for other content. In both small caches (1GB) and large caches (1TB), this redundancy means less content diversity, reducing the likelihood of cache hits for less-frequently requested segments. Although, LCE gets a high ratio of redundancy that is logically leads to increase average hit ratio. Li et al. (2016) highlight that caching without coordination or awareness, like LCE, is fundamentally limited in achieving high cache efficiency, regardless of capacity [334].

This increasing of hit happen when request the same request, but the scenario of getting different request increase the miss ratio due to the redundancy, large-scale of video, and the memory is occupied for all routers with same video.

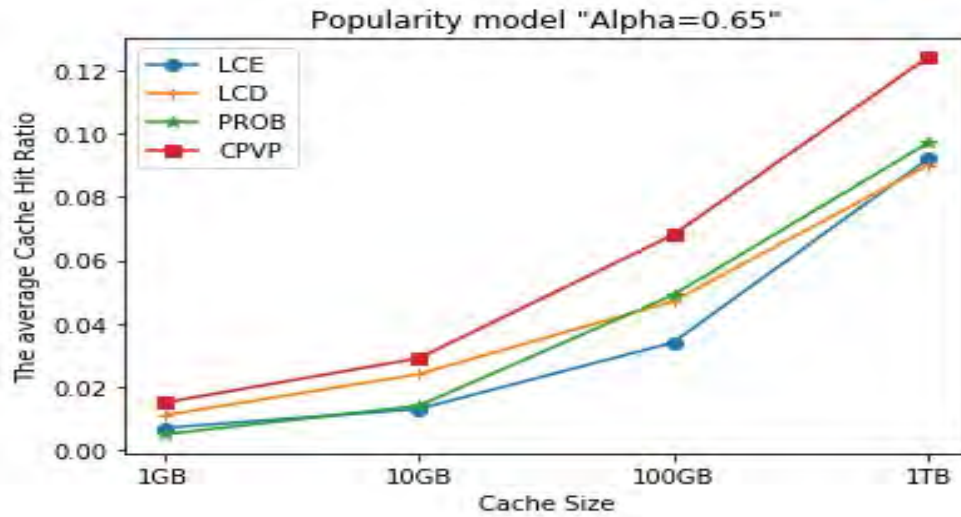


Figure 6.4. Average Cache Hit Ratio Using DTelekom Topology

However, in DTelekom topology, the average cache hit ratio is also larger than the other three LCE, LCD, and Prob along 1GB to 1TB. The results of LCD are higher in 1GB, but its result is similar to Prob in 100GB and similar to LCE in 1TB. Along the result, LCE register the worse mechanism compared to CPVP, LCD, and Prob caching placement mechanisms, as shown in Figure 6.4.

The Table 6.1, shows the results of average cache hit ratio to evaluate CPVP mechanism by comparing with three mechanisms (LCE, LCD, and Prob) in different scenarios of multi cache size (1GB, 10GB, 100GB, and 1TB) and various four topologies (Tree, Abilene, GEANT, and DTelekom). In these different scenario, the results show that CPVP mechanism is the best due to it's diversity. While LCE is the worst in terms of average cache hit ratio due to it's redundancy [332, 333]. In addition to that the results is enhanced while the cache size is increased continuously. However, in the new mechanism (CPVP) each request will have probability to get more hit  $Hit_r$

**Table 6.1**

The Average Cache Hit Ratio for Four Placement Mechanisms Using 1 GB to 1 TB

Cache Size	Tree Topology				Abilene Topology				GEANT Topology				DTelekom Topology			
	LCE	LCD	Prob	CPVP	LCE	LCD	Prob	CPVP	LCE	LCD	Prob	CPVP	LCE	LCD	Prob	CPVP
<b>1GB</b>	0.009	0.013	0.007	0.018	0.006	0.09	0.004	0.013	0.005	0.007	0.003	0.011	0.007	0.011	0.005	0.015
<b>10GB</b>	0.016	0.027	0.017	0.032	0.011	0.023	0.013	0.028	0.009	0.02	0.011	0.027	0.013	0.024	0.014	0.029
<b>100GB</b>	0.035	0.048	0.052	0.072	0.033	0.046	0.047	0.066	0.031	0.044	0.045	0.064	0.034	0.047	0.049	0.068
<b>1TB</b>	0.094	0.092	0.101	0.129	0.89	0.088	0.095	0.123	0.086	0.085	0.092	0.199	0.092	0.09	0.097	0.124
<b>Average</b>	0.0385	0.045	0.044	0.0628	0.0348	0.0415	0.0398	0.0575	0.0327	0.039	0.0378	0.0753	0.0365	0.043	0.0413	0.059
<b>CPVP is % Best</b>	<b>38.6%</b>	<b>28.3%</b>	<b>29.5%</b>	<b>39.6%</b>	<b>27.8%</b>	<b>30.9%</b>	<b>49.8%</b>	<b>56.5%</b>	<b>48.2%</b>	<b>49.8%</b>	<b>38.1%</b>	<b>30.1%</b>	<b>38.1%</b>	<b>27.1%</b>	<b>30.1%</b>	<b>30.1%</b>

due to partitioning the file in three routers. Data will be near the three routers by minimizing hop distance.

Finally, for easier finding the percentage of enhancement, the average of results for various cache sizes is dependent to get the enhancement percentage of the CPVP mechanism by comparing it with three placement mechanisms. Therefore, the enhanced percentage of the CPVP mechanism is so clearly in the Table 6.1 in terms of an average cache hit ratio by different topologies. Whereas, it is better than LCE by (38.6%, 39.6%, 56.5%, and 38.1%) in different topologies (Tree, Abilene, GEANT, and DTelekom) respectively. Thus, CPVP gets a higher percentage of enhancement than LCE by 56.5% by using the GEANT topology. While, CPVP gets better than LCD by 28.3%, 27.8%, 48.2%, and 27.1% by using Tree, Abilene, GEANT, and DTelekom in sequence. Psaras et al. [16] state that LCE creates excessive redundancy and underutilises cache capacity, lowering the effective cache hit ratio despite its aggressive behavior. Finally, CPVP is getting the best results with an average hit ratio of 29.5%, 30.9%, 49.8%, and 30.1% respectively. The conclusion is that the highest enhancement results are getting with the GEANT topology for all three mechanisms above.

Generally, The CPVP is got the highest enhancement performance 56.5% from LCE in terms of the average hit ratio by using GEANT topology. While it registered the lowest enhancement performance of 27.1% from LCD in terms of the average hit ratio by using DTelekom topology.

### **6.2.2 Average E2EDelay by Using Four Topologies**

The E2EDelay presents the whole latency required to cross the network from the client (source) to the destination. It is the most significant metric which is necessary to evaluate VoD performance in the network. It is used to evaluate the CPVP mechanism

by ndnSIM2.0 simulation and various cache sizes (1GB, 10GB, 100GB, and 1TB) by using four topologies: Tree, Abilene, GEANT, and DTelekom as shown in the Figures. At the same time, the size for each file is taken randomly from the dataset (SVC-DASH).

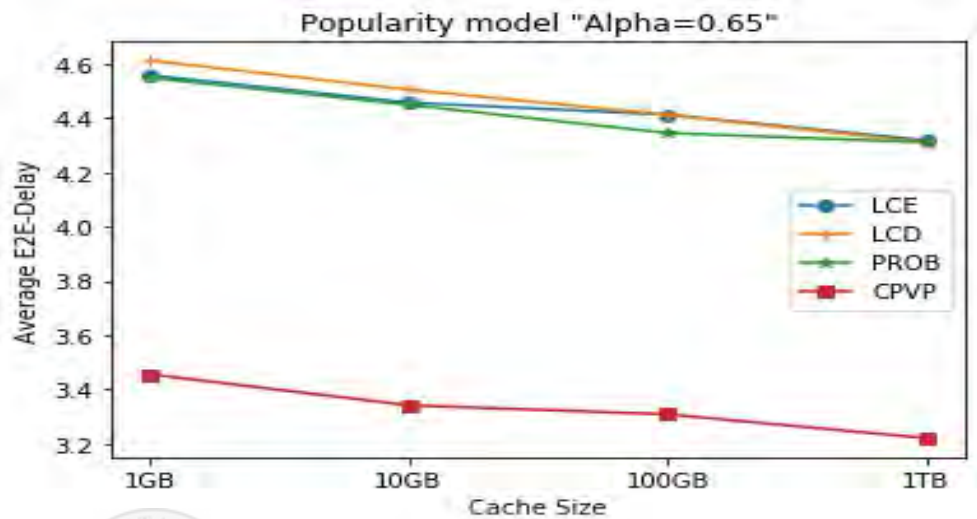


Figure 6.5. Average E2EDelay Using Tree Topology

In Figure 6.5, the results of E2EDelay are presented. We can see that all three previous mechanisms get the nearest results to each other when comparing them with the new mechanism CPVP. CPVP demonstrates superior performance across all cache sizes, especially in larger caches like 1TB, due to its ability to optimally cache partitioning content closer to users.

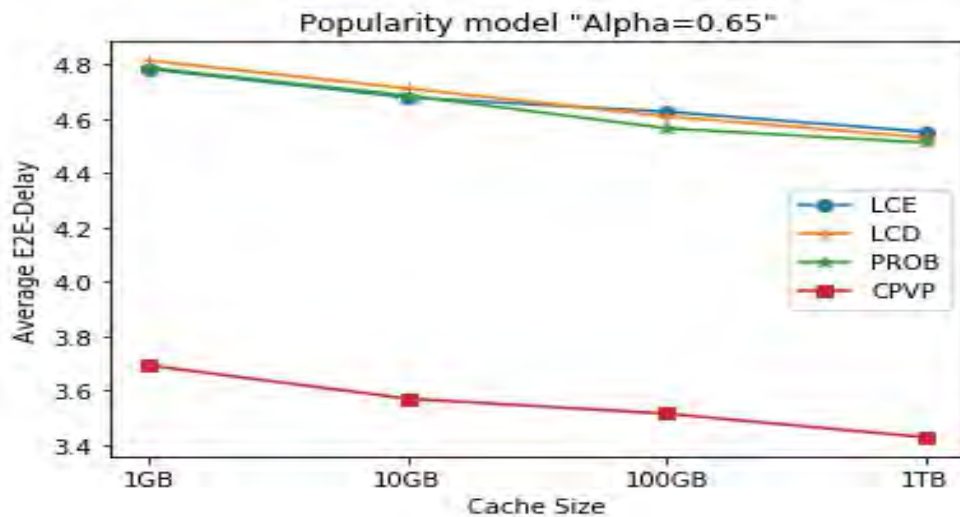


Figure 6.6. Average E2EDelay Using Abilene Topology

The E2EDelay is decreased when the cache size is increased for all four mechanisms (LCE, LCD, Prob, and CPVP). In the Abilene topology, there is a competition between Prob and LCD about the second level for delayed performance but Prob is better than LCD generally. Whatever the first level is registered to the CPVP mechanism in Figure 6.6.

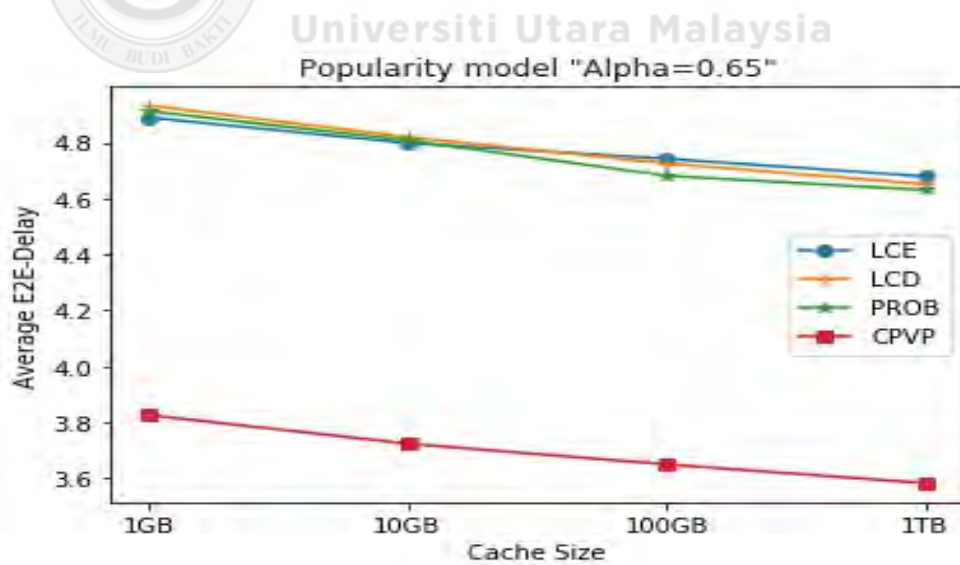


Figure 6.7. Average E2EDelay Using GEANT Topology

In the GEANT topology, Figure 6.7 referred to that CPVP is still registered the best compared to the others for  $\alpha = 0.65$ .

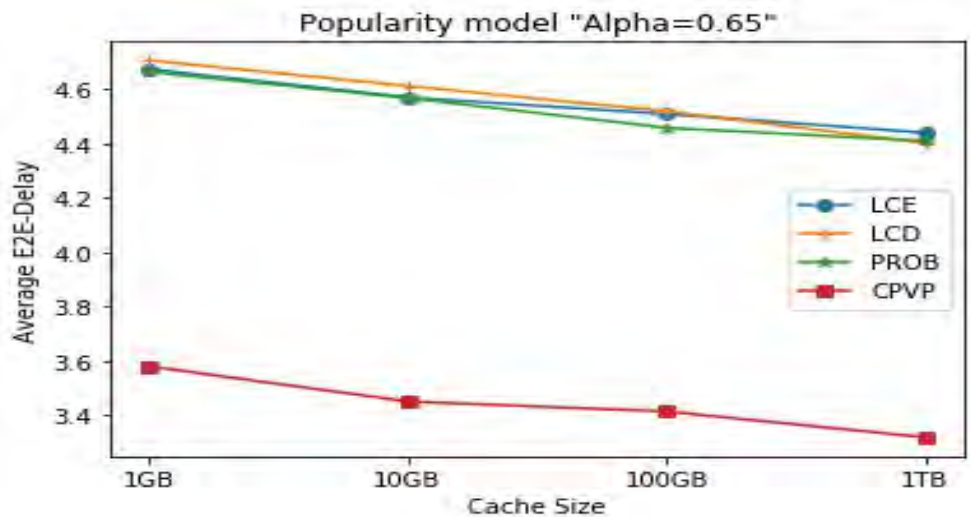


Figure 6.8. Average E2EDelay Using DTelekom Topology

According to Figure 6.8, the results of E2EDelay are got more significant result than the other. It is very important to the user.

NDN performance metrics determine the E2EDelay path by adding the node delay at each node and the link delay at each link along the route. The protocol processing time and the queuing delay at node  $i$  for the connection  $(i, j)$  are both included in node delay. Considering a wireless link, it is given that the propagation delays are almost equal for every hop on the path [282, 292], [292, 283]. In Table 6.2, still the CPVP mechanism register the best mechanisms in all cache sizes and all four topologies due to its design to cache in the nearest three routers to the consumers.

Lastly, the average of performance results for various cache sizes is dependent to get the enhancement percentage of the CPVP mechanism by comparing it with three placement mechanisms (LCE, LCD, and Prob). Therefore, the enhanced percentage of the CPVP mechanism is so clearly in the Table 6.2 in terms of E2EDelay by different topologies. Whereas, it is better than LCE by (24.9%, 23.8%, 22.7% , and 24.3%) in different topologies (Tree, Abilene, GEANT, and DTelekom) respectively. Thus,

**Table 6.2**  
Average E2EDelay for Four Placement Mechanisms Using 1GB to 1TB

Cache Size	Tree Topology			Abilene Topology			GEANT Topology			DTelekom Topology						
	LCE	LCD	Prob	LCE	LCD	Prob	LCE	LCD	Prob	LCE	LCD	Prob	LCE	LCD	Prob	
<b>1GB</b>	4.558	4.612	4.549	4.782	4.813	4.785	4.889	4.932	4.912	3.825	4.932	4.819	4.673	4.932	4.819	3.579
<b>10GB</b>	4.457	4.505	4.449	4.677	4.71	4.684	4.798	4.819	4.808	3.723	4.611	4.567	4.569	4.611	4.567	3.45
<b>100GB</b>	4.413	4.412	4.345	4.625	4.608	4.564	4.742	4.725	4.681	3.649	4.518	4.457	4.508	4.518	4.457	3.413
<b>1TB</b>	4.318	4.31	4.311	4.551	4.53	4.511	4.679	4.651	4.63	3.582	4.401	4.409	4.439	4.401	4.409	3.318
<b>Average</b>	4.437	4.459	4.414	4.659	4.665	4.636	4.777	4.782	4.76	3.695	4.616	4.563	4.547	4.616	4.563	3.44
<b>CPVP is %Best</b>	<b>24.9%</b>	<b>25.3%</b>	<b>24.5%</b>	<b>23.8%</b>	<b>23.9%</b>	<b>23.4%</b>	<b>22.7%</b>	<b>22.7%</b>	<b>22.3%</b>	<b>24.3%</b>	<b>25.5%</b>	<b>24.6%</b>	<b>24.3%</b>	<b>25.5%</b>	<b>24.6%</b>	

CPVP gets a higher percentage of enhancement than LCE by 24.9% by using the Tree topology. While, CPVP gets better than LCD by 25.3%, 23.9%, 22.7%, and 25.5% by using Tree, Abilene, GEANT, and DTelekom in sequence. Finally, CPVP is getting the best results with E2EDelay of 24.5%, 23.4%, 22.3%, and 24.6% respectively. These results are help to enhance QoE because E2EDelay refers to the total time taken for a content request to be satisfied when the interest is sent to when the corresponding data packet is received [335]. The conclusion is that the highest enhancement of E2EDelay results are getting with the Tree topology for all three mechanisms above.

Generally, The CPVP is registered the highest enhancement performance 25.5% from LCD in terms of the E2EDelay by using DTelekom topology. While it registered the lowest enhancement performance of 22.3% from Prob in terms of the total reduction in network footprint by using GEANT topology.

### **6.2.3 Server Load Reduction by Using Four Topologies**

Server load reduction is the metric that refers to how much the server load is decreased on the origin servers. Server load reduction is an essential in modern networking and computing to improve efficiency, reduce costs, and ensure that services can scale effectively without overwhelming server resources. Server load reduction is a more practical metric than cache hit ratio in VoD deployment contexts [336, 337]. The Figures reflect the results of this metric to evaluate the CPVP mechanism.

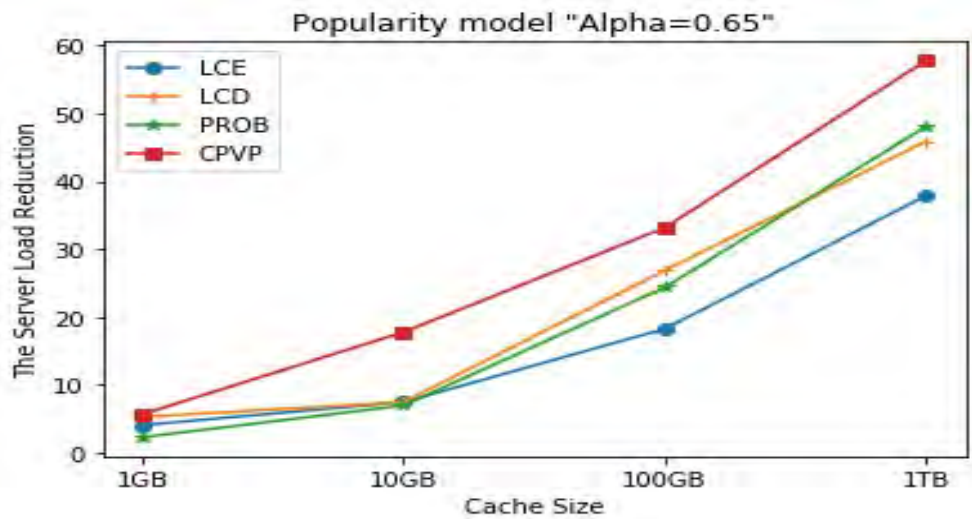


Figure 6.9. Server Load Reduction Using Tree Topology

Figure 6.9 refers to how much the reduction in server load is by implementing the CPVP mechanism using Tree topology,  $\alpha = 0.65$ , and comparing it with LCE, LCD, and Prob mechanisms. Finally, CPVP got significant results making it the best mechanism.

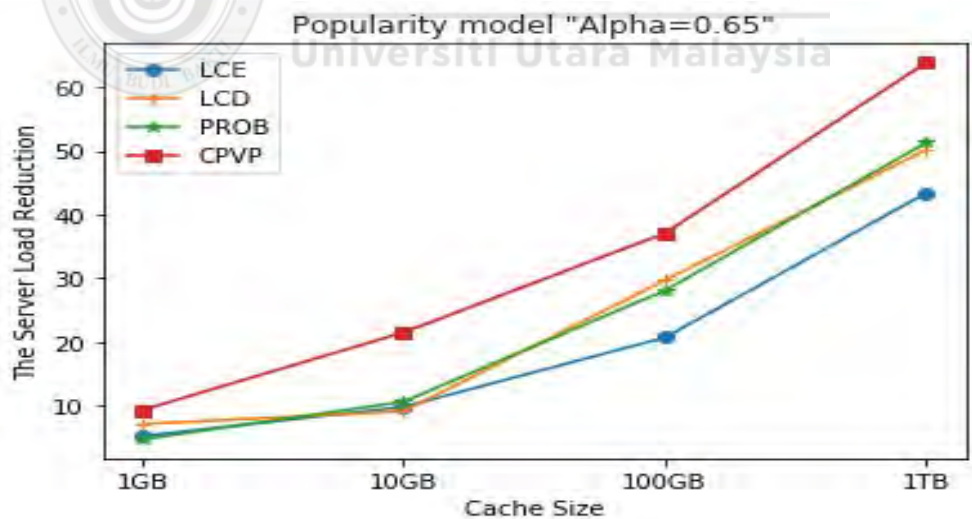


Figure 6.10. Server Load Reduction Using Abilene Topology

Figure 6.10 represents the server load reduction using Abilene topology to conclude that LCE is the worst mechanism while CPVP is the best one. At the same time, there is a competition between LCD and Prob mechanisms to get the middle level of results.

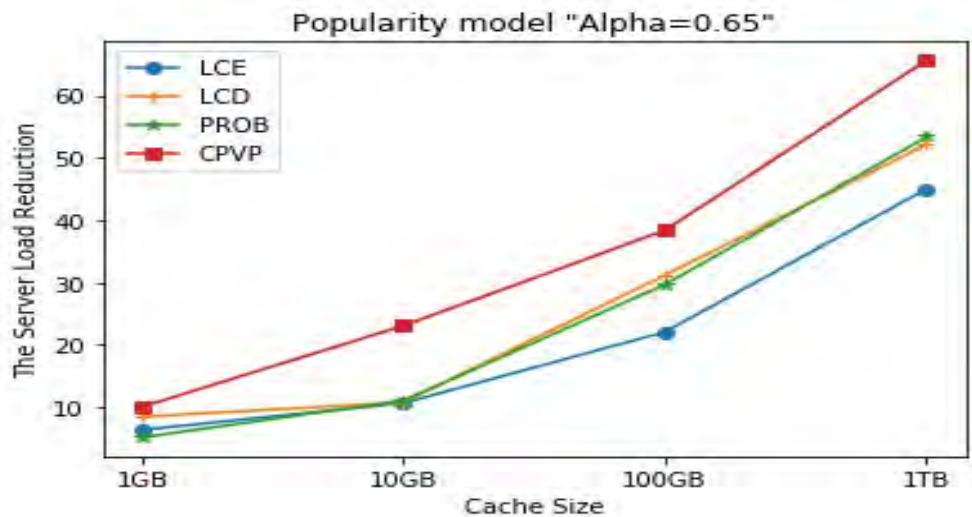


Figure 6.11. Server Load Reduction Using GEANT Topology

Figure 6.10 displays the outcomes of the server load reduction using the GEANT topology. The relation between the results of these four topologies it looked the same in Figure 6.10 with different values.

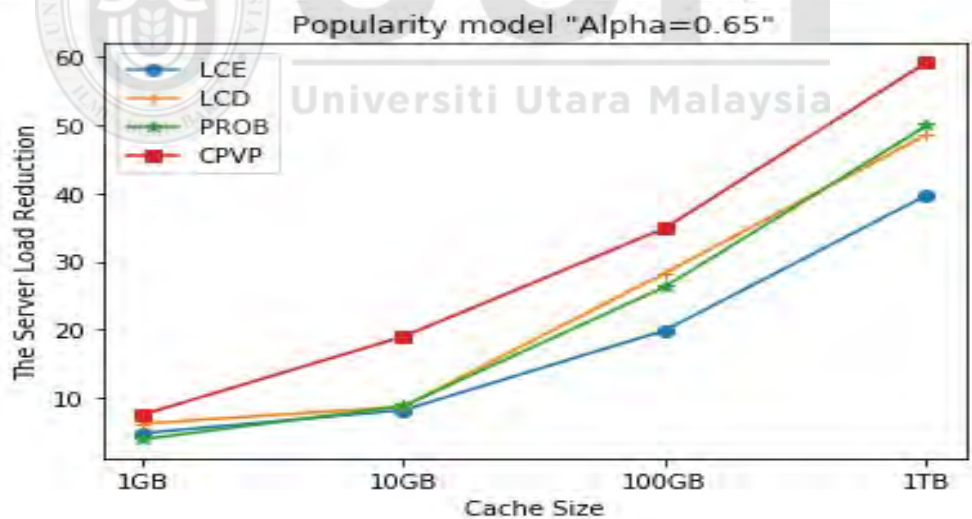


Figure 6.12. Server Load Reduction Using DTelekom Topology

Figure 6.12, the results reflect that the CPVP mechanism is the best, while the LCE mechanism is the worst when comparing the results of them to these four mechanisms (LCE, LCD, CPVP, and Prob).

**Table 6.3**  
Server Load Reduction for Four Placement Mechanisms Using 1GB to 1TB

Cache Size	Tree Topology				Abilene Topology				GEANT Topology				DTelekom Topology			
	LCE	LCD	Prob	CPVP	LCE	LCD	Prob	CPVP	LCE	LCD	Prob	CPVP	LCE	LCD	Prob	CPVP
<b>1GB</b>	4.1	5.3	2.3	5.7	5.2	7.1	4.7	9.3	6.4	8.5	5.2	10.1	4.8	6.2	3.9	7.5
<b>10GB</b>	7.5	7.5	7	17.8	9.8	9.1	10.6	21.5	10.7	10.8	11.1	23.1	8.2	8.7	8.8	19
<b>100GB</b>	18.3	26.9	24.4	33.2	20.7	29.7	28.1	37.1	22.1	31.2	29.7	38.4	19.9	28.3	26.3	34.9
<b>1TB</b>	37.9	45.9	48.1	57.8	43.4	50.2	51.3	63.9	45	52.1	53.3	65.6	39.7	48.6	49.9	59.2
<b>Average</b>	16.95	21.4	20.45	28.625	19.78	24.03	23.68	32.95	21.05	25.65	24.83	34.3	18.15	22.95	22.23	30.15
<b>CPVP is % Best</b>	<b>40.8%</b>	<b>25.2%</b>	<b>28.6%</b>		<b>40%</b>	<b>27.1%</b>	<b>28.1%</b>		<b>38.6%</b>	<b>25.2%</b>	<b>27.6%</b>		<b>39.8%</b>	<b>23.9%</b>	<b>26.3%</b>	

Finally, the Table 6.3 is summarized all the results of server load reduction to four mechanisms with four topologies and various cache sizes with  $\alpha = 0.65$ . When notifying all the scenarios above, concluding that LCE is the worst in terms of server load reduction, the CPVP mechanism is the best. In addition that this performance metric is enhanced when the cache size is increased.

However, server load reduction reflects the reasons for giving significant results. This is because server load represents the number of requested video which needs to be served from the server [338]. The CPVP mechanism is very significant in serving the request by intermediate cache before arriving the request to the server.

Lastly, due to easier getting the enhancement percentage, the average results for all cache sizes are based to get the enhancement percentage of the CPVP mechanism by comparing it with the three placement mechanisms. Therefore, the enhanced percentage of the CPVP mechanism is very significant in Table 6.3 in terms of the server load reduction by different topologies. Whereas, it is better than LCE by (40.8%, 40%, 38.6%, and 39.8%) in various topologies (Tree, Abilene, GEANT, and DTelekom) respectively. Thus, CPVP gets a higher percentage of enhancement than LCE by 40.8% by using the Tree topology. While, the CPVP gets better than LCD by 25.2%, 27.1%, 25.2%, and 23.9% by using Tree, Abilene, GEANT, and DTelekom in sequence. Finally, CPVP is getting the best results with server load reduction of 28.6%, 28.1%, 27.6%, and 26.3% respectively. The conclusion is that the highest enhancement results are getting with the Tree topology for all three mechanisms above.

Generally, The CPVP got the highest enhancement performance 40.8% from LCE in terms of the server load reduction by using GEANT topology that helps to enhance QoE [339]. While it registered the lowest enhancement performance of 25.2% from LCD in terms of the server load reduction by using two topology: Tree topology, and

GEANT topology.

#### 6.2.4 Total Reduction in the Network Footprint by Using Four Topologies

In the context of NDN for VoD, total reduction in the network footprint is the outcome of intelligent caching, optimised request aggregation, and efficient content delivery, all of which work together to reduce bandwidth consumption, limit redundant data transfers, and localize content retrieval. This approach minimises the overall impact on network resources, allowing for more scalable, efficient, and sustainable video distribution over the NDN infrastructure. The four figures illustrate the use of this performance metric to evaluate the CPVP mechanism.

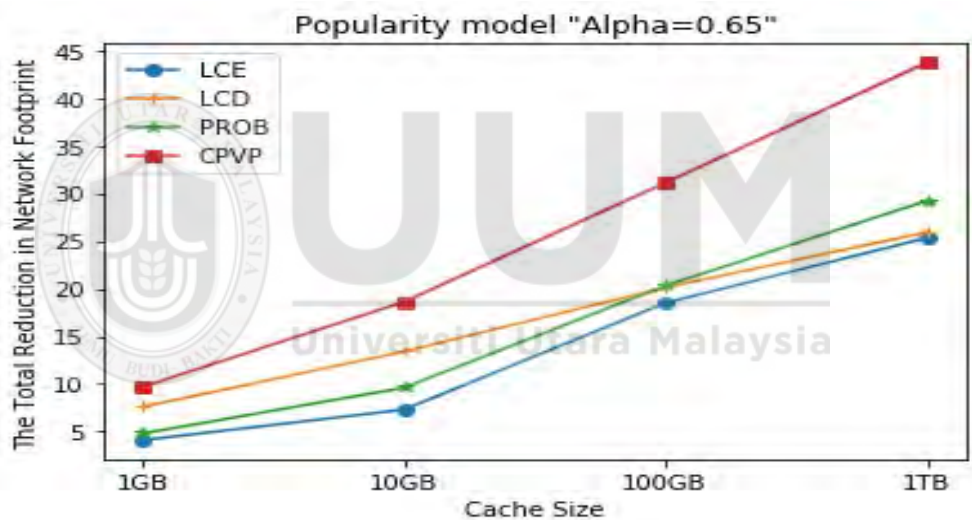


Figure 6.13. Total Reduction in the Network Footprint Using Tree Topology

This Figure 6.13 shows that CPVP mechanism gives significant results in terms of the total reduction in network footprint along various cache sizes. Prob mechanism have the second level after CPVP in 1TB. LCD performs well in smaller caches 1GB and 10GB, where its strategy of caching close to the requester helps reduce network traffic, but it is outperformed in larger caches where more sophisticated caching strategies (like CPVP and Prob) can better optimise the footprint.

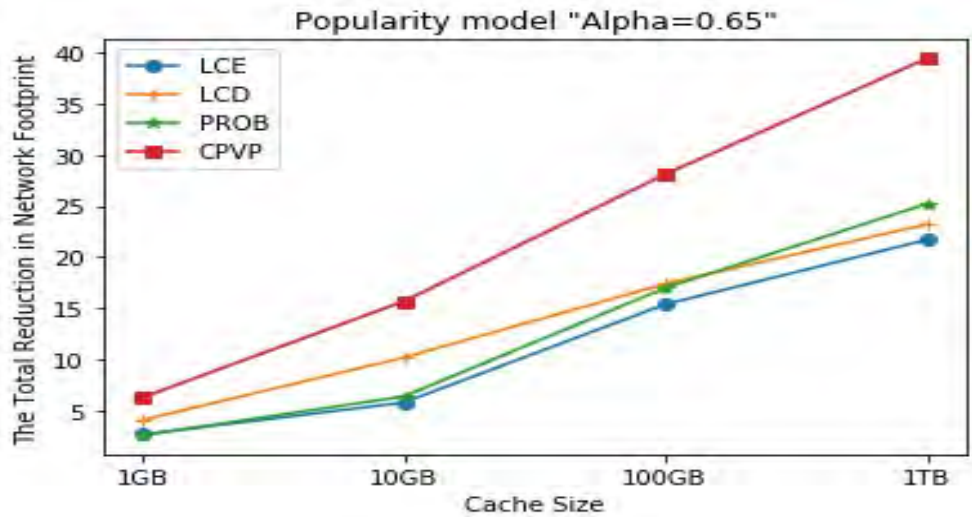


Figure 6.14. Total Reduction in the Network Footprint Using Abilene Topology

However, the Figure 6.14 is similar to the Figure 6.14 above, when comparing the results of these four mechanism except the highest value in this Figure is less than the Figure before.

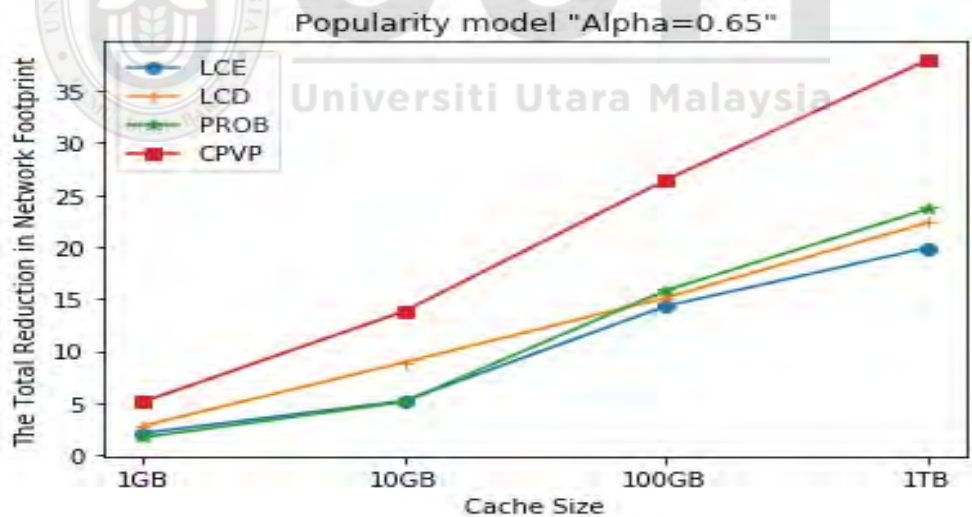


Figure 6.15. Total Reduction in the Network Footprint Using GEANT Topology

This performance metric by using GEANT topology is gave the best result by comparing it to other three topologies. In addition to that CPVP mechanism is the best and LCE mechanism is the worst.

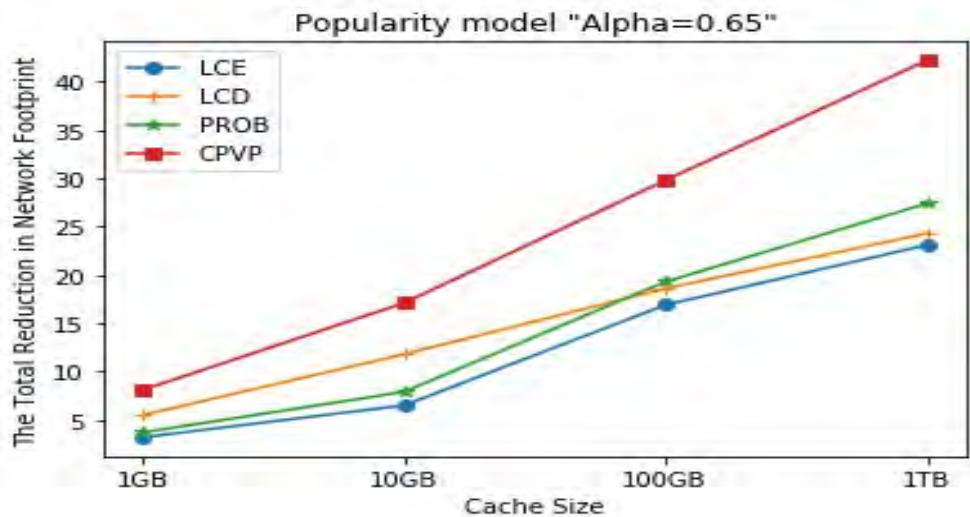


Figure 6.16. Total Reduction in the Network Footprint Using DTelekom Topology

In addition to that the results in DTelekom topology as shown in Figure 6.16 is similar to Figure 6.15 of this metrics.

The Table 6.4 contains all the results of this metrics to four caching strategies, four cache sizes, and Four topologies to come out with comprehensive analysis and to evaluate the CPVP mechanism. The Finding is the CPVP mechanism is the best, LCE mechanism is the worst, in addition to that the results is enhanced while increasing the cache sizes. The CPVP mechanism is minimised the number of hops which taken by the request to get a hit, because it caches the video content in the nearest three neighbour routers.

Lastly, the average performance results for different cache sizes are dependent to get the enhancement percentage of the CPVP mechanism by comparing it with three

placement mechanisms (LCE, LCD, and Prob). Therefore, the enhanced percentage of the CPVP mechanism is so clearly in the Table 6.4 in terms of the total reduction in network footprint by different topologies. Whereas, it is better than LCE by

**Table 6.4**  
Total Reduction in Network Footprint for Four Placement Mechanisms Using 1GB to 1TB

Cache Size	Tree Topology				Abilene Topology				GEANT Topology				DTelekom Topology			
	LCE	LCD	Prob	CPVP	LCE	LCD	Prob	CPVP	LCE	LCD	Prob	CPVP	LCE	LCD	Prob	CPVP
<b>1GB</b>	4.1	7.6	4.8	9.7	2.7	4.1	2.6	6.3	2.1	2.8	1.7	5.1	3.2	5.5	3.7	8.1
<b>10GB</b>	7.3	13.4	9.6	18.6	5.8	10.2	6.4	15.7	5.2	8.9	5.1	13.8	6.5	11.8	7.9	17.1
<b>100GB</b>	18.5	20.1	20.3	31.1	15.4	17.4	17	28.1	14.3	15.1	15.8	26.4	16.9	18.6	19.3	29.8
<b>1TB</b>	25.3	25.9	29.2	43.7	21.7	23.2	25.2	39.4	19.9	22.3	23.6	37.9	23.1	24.3	27.4	42.2
<b>Average</b>	13.8	16.75	15.98	25.78	11.4	13.725	12.8	22.375	10.375	12.275	11.55	20.8	12.425	15.05	14.58	24.3
<b>CPVP is % Best</b>	<b>64.5%</b>	<b>35%</b>	<b>38%</b>		<b>49.1%</b>	<b>38.7%</b>	<b>42.8%</b>		<b>50.1%</b>	<b>41%</b>	<b>44.5%</b>		<b>48.9%</b>	<b>38.1%</b>	<b>40%</b>	

(46.5%, 49.1%, 50.1%, and 48.9%) in different topologies (Tree, Abilene, GEANT, and DTelekom) respectively. Thus, the CPVP gets a higher percentage of enhancement than LCE by 50.1% by using the GEANT topology. While, the CPVP gets better than LCD by 35%, 38.7%, 41%, and 38.1% by using Tree, Abilene, GEANT, and DTelekom sequentially. Finally, the CPVP is getting the best results with a total reduction in network footprint of 38%, 42.8%, 44.5%, and 40% respectively. The conclusion is that the highest enhancement of total reduction in network footprint results is getting with the GEANT topology for all three mechanisms.

Generally, The CPVP got the highest enhancement performance 50.1% from LCE in terms of the total reduction in network footprint by using GEANT topology because CPVP reduces the cumulative hop count for content delivery by over half compared to LCE, especially in a large-scale network like GEANT. CPVP achieves higher cache diversity and locality, ensuring that the content is not over-replicated across nodes, the right content is cached at the right location because the behavior-aware caching mechanisms lower total footprint by localizing content near demand customer [340]. While it registered the lowest enhancement performance of 35% from LCD in terms of the total reduction in network footprint by using Tree topology.

### **6.3 The Enhancement Percentage of the CPVP Mechanism**

CPVP registered the highest percentage of enhancement as 56.5% from LCE using the GEANT topology. While It registered the lowest percentage of enhancement as 22.3% from Prob using GEANT topology as shown in Table 6.5. CPVP is registering the best results in four performance metrics using four different topologies due to its diversity, caching, and caching one time nearest to the consumer because it satisfies all functional requirements, which are obtained from the first objective. CPVP achieves the highest network footprint reduction on the GEANT topology because it coordinates

cache placements across nodes to avoid duplication, and prepositions content close to regional users, in addition to it matches the structural complexity and regional demand diversity inherent in GEANT. Raaid et al. [106] emphasized that GEANT topology is utilised to test the significance of cache management selection strategies. As a result, it minimises hop counts, shortens retrieval paths, and significantly reduces the overall network traffic load compared to other mechanisms.

**Table 6.5**

Summary of Enhancement Percentage of the CPVP Mechanism

Metrics	Highest			Lowest		
	Percentage	Placement Mechanism	Topology	Percentage	Placement Mechanism	Topology
Average Cache Hit Ratio	56.5%	LCE	GEANT	27.1%	LCD	DTelekom
Average E2EDelay	25.5%	LCD	DTelekom	22.3%	Prob	GEANT
Server Load Reduction	40.8%	LCE	GEANT	25.2%	LCD	GEANT, Tree
Total Reduction in Network Footprint	50.1%	LCE	GEANT	35%	LCD	Tree

## 6.4 Summary

This chapter looked at how well the CPVP mechanism works compared to the NDN-specific LCE, LCD, and Prob caching methods. The new mechanism was accomplished using the ndnSIM 2.0 simulator for the evaluation of average cache hit ratio, average E2EDelay, server load reduction, and total reduction in network footprint. Finally, the CPVP mechanism achieves the best performance results by comparing LCE, LCD, and Prob during VoD in NDN. It registered the best results in four performance metrics using four different topologies due to its diversity, caching and caching once nearest to the consumer. CPVP is satisfied with all functional requirements that were obtained from the first objective. However, CPVP has three

schemes: a VAS to ensure efficient aggregation and retrieval of video content, a VCDS to optimise caching decisions to minimise redundancy and delay, and VPS to dynamically partition video files to enhance cache utilisation and retrieval speed. These three schemes cooperate by using cooperative tables, dynamic partitioning, and coordinated caching decisions that contribute to its superior performance. Thus, the QoE is developed. As a result, the CPVP mechanism demonstrated that it might be an ideal option for caching mechanisms during VoD over the Internet in the future. Thus, the limitation of the CPVP mechanism is especially to deal with large-scale content to serve VoD. While the future work of it is such as testing CPVP under more complex network scenarios or integrating additional performance metrics.



## CHAPTER SEVEN

### CONCLUSION AND FUTURE WORKS

The Internet's rapid expansion has disrupted communication paradigms with Named Data Networking (NDN) and online video storage. The study's highlighted concern is the excessive delivery times associated with the Video on Demand (VoD) burden brought on by consumers' demands for online videos. Furthermore, the Internet's quick expansion and high video storage require that it be saved in several packets, which takes a lot of time. In **Chapter Two**, we compared and analysed nine mechanisms of cache placement in NDN while the VoD was operating. Furthermore, according to the empirical simulation results, using the simulation empirically of seven mechanisms in **Chapter Four**, the Probabilistic cache placement mechanism (Prob) is the optimum placement mechanism for the delivery time. This study concludes that the NDN has to be enhanced by incorporating elements of various cache placement mechanisms.

Designing a Cooperated Placement by method for Named Data Networking (NDN) is the primary objective of this thesis. This research achieves many challenges in caching placement contributions NDN to implement seven mechanisms for caching placement with VoD. The literature analyses seven placement caching mechanisms with VoD in NDN. Then pay attention to the methodology phases and variables like topologies and popularity models. The simulation settings were chosen with care. The performance of the CPVP mechanism was extensively evaluated through the ndnSIM 2.0 simulator. In Chapter Four, we discussed the empirical evaluation of seven placement mechanisms (LCE, LCD, Prob, Pprob, Rand, Centrality, and Cross) to identify the features of these mechanisms during VoD. While the validation of the new placement mechanism (CPVP) is tested using four topologies: Tree, Abilene, GEANT, and DTelekom. The cache size ranged from 1GB to 1TB, while the catalogue size was in elements. Two

Zipf model values are chosen for popularity, i.e., 0.65 and 2.0 in Chapter Four, but 0.65 is used to design the new one because CPVP does not depend on the Zipf model. The remainder of this chapter is organised as follows:

Section 7.1 discusses the significance of the **CPVP** mechanism for developing the Internet in the future and its potential implementation. At the same time, the limits are indicated in Section 7.3, and the research contributions are given in Section 7.2. Eventually, Section 7.4 offers some recommendations for more research.

## 7.1 Research Summary

As described in **Chapter One**, the NDN network is categorized as a multi-stage caching system, with demands from geographically distributed nodes and the caches being physically spread out over the network. The request is now locally fulfilled if the requested material is already present somewhere and has been cached by a network node. As a result, the issue of efficient caching systems for content placement during VoD demand arises. In light of this, the main goal of this thesis was to introduce a new caching mechanism for content placement.

Even though several mechanisms have been developed for this purpose, as stated in **Chapter Two**, it was discovered and determined that the current solutions (LCE, LCD, Prob, Pprob, Cross, Rand, and Centrality) are unable to address the difficulties posed by content placement during VoD. For instance, the default-ICN mechanism, LCE, makes the material more redundant. Coordination Caching Placement by Video Partitioning (CPVP), a new mechanism for NDN, was created due to this research study.

To fulfil the goals of this study, the Research methodology of the operational framework, conceptual model, verification and validation was adopted and discussed in

**Chapter Three.** However, in **Chapter Four**, empirical evaluation for seven placement caching mechanisms is achieved by using the ndnSIM 2.0 simulator on four topologies (Tree, Abilene, GEANT, and DTelekom) with various cache sizes from 1GB to 1TB and Zipf parameters equal to 0.65 or 2.0. It can be said that the Prob mechanism had a minor time delay and performed well according to the performance parameters that were looked at. The Centrality placement mechanism is the worst one in all performance metrics. In addition, the results of all seven mechanisms are best when the cache is increased. At the same time, the results are enhanced for all seven placement mechanisms when the Zipf distribution is increased from  $\alpha = 0.65$  to  $\alpha = 2.0$ .

In **Chapter Five**, the theoretical analysis of the placement problem and the analysis of the coordination and non-coordination placement conclude that coordinated placement is the best during VoD services. The CPVP mechanism is the new placement mechanism designed. It contains three schemes: the Video Aggregation Scheme (VAS) to collect the content, the Video Cache Decision Scheme (VCDS) to decide whether to cache or not and where to cache the content, and the Video Partitioning Scheme (VPS) to divide the content of the video file into three blocks and cache these three blocks in the nearest three neighbour routers to the edge.

In **Chapter Six**, the simulation results showed that the CPVP approach raises the average cache hit rate, lowers server load, and shrinks the size of the entire network. Positive effects result from the average End-to-End Delay (E2EDelay) being reduced considerably. The simulation findings highlight that in performing the performance evaluation of the **CPVP** mechanism, the primary goals of this work have been entirely met. Generally, it registered the highest enhancement percentage at 56.5% of LCE using the GEANT topology. While it registered the lowest percentage of enhancement at 22.3% of Prob using the GEANT topology.

## 7.2 Research Contribution

By combining the prior research on NDN and VoD, this study will favour VoD users and publishers, leading to greatly better academic learning. The main contribution of this study was to design a **CPVP** mechanism to enhance the performance of the future Internet. Moreover, three schemes, the **VAS**, **VCDS**, and **VPS**, are proposed for achieving better results for the CPVP mechanism. The specific contributions consist of the following points:

1. The adaptation of a new mechanism (**CPVP**) was produced for content placement by partitioning, leading to improved caching gain and simplifying content access by creating three new schemes:
  - a) Video Aggregation Scheme (**VAS**) is designed to collect interesting data.
  - b) Video Cache Decision Scheme (**VCDS**) is designed to decide whether to cache or not. If it decides to cache, it chooses which router to cache and which block of this content will be cached in this specific router.
  - c) Video Partitioning Scheme (**VPS**) decides how many chunks need to be cached in each of the three routers.
2. The design of the **CPVP** mechanism can be used as a benchmark for any planned advancements in future NDN caching.
3. The analysis of the empirical evaluation of seven placement cache mechanisms.
4. The development of a theoretical model to analyse the caching placement problem.
5. The development of a theoretical model to analyse the coordination and non-coordination placement mechanisms.
6. The development of cooperative tables, which are:
  - a) Cooperative Interest Chunk Table (CIC).
  - b) Cooperative Data Chunk Table (CDC).

- c) Cooperative Content Store Table (CCS)
  - d) Cooperative Pending Interest Table (CPIT)
7. The development of a new NDN caching system for video-on-demand applications results in the following contributions that are relevant:
- a) Raising the average cache hit proportion of the requested content items, making it more effective for data queries of frequently visited objects.
  - b) Reducing average E2EDelay for the results of content placement, leading to better memory management.
  - c) Minimising the server load and increasing hop reduction in the network footprint.
  - d) Enhancing the VoD services, which directly affect enhancing the Quality of Services (QoS).
8. The implementation of the **CPVP** mechanism in ndnSIM2.0 and comparison of its outcomes with other caching placement mechanisms (LCD, LCE, and Prob) by four topologies (Tree, Abilene, GEANT, and DTelekom) in terms of the server load reduction, average cache hit ratio, E2EDelay, and decrease in network footprint allowed for the verification and validation of the CPVP mechanism.
9. Enhancing VoD services in academic settings can transform the learning experience by providing greater accessibility, personalisation, and engagement. It supports diverse learning styles, facilitates collaboration, and offers scalable, high-quality educational content. By leveraging these benefits, educational institutions can create a more effective, inclusive, and flexible learning environment for students.

### 7.3 Research Limitations

Despite being well-planned and incorporating various guiding principles and methods, this research remains limited to a few specific applications. Crucially, the lack of a specially designed NDN processor could potentially alter the simulation results when applied to actual NDN processors. In ndnSIM 2.0, packet transmission and reception rely on IP processors.

The suggested approach also works effectively in fixed wireless networks with infrastructure. However, applying the proposed approach to every ad hoc network, as in [159], will complicate the task of identifying the router with the highest number of outgoing interfaces. Also, the new system is designed so that all three blocks can be cached at the same time. This is possible because content is replaced based on "Last Recently Used" (LRU), which doesn't affect the overall result. Additionally, we specifically designed the new mechanism, CPVP, for large-scale video services like VoD.

### 7.4 Future Works

The new mechanism in this study aims to improve the performance of NDN during VoD services by using various scenarios. However, we can pursue certain limitations and unresolved issues in new directions. Current progress allows for the exploration of many more directions. This section outlines some possible extensions.

Additionally, it would be intriguing to enhance the CPVP mechanism so that it can handle all sorts of interest requests and serve as the optimum placement mechanism for practically any material, including video files, in addition to designing a new replacement mechanism, especially with the CPVP placement mechanism to get sufficient storage for caching. This results in optimal service for both the consumer and

the producer, who are not satisfied with the data size. Thus, the quality of service (QoS) will be enhanced to ensure the Quality of Experience (QoE) to serve the consumers.

The effectiveness of CPVP was evaluated in comparison to the default placement mechanism (LCE) of NDN, the Prob placement mechanism [16], and LCD (which obtained less redundancy than the LCE). In various network scenarios, such as cloud infrastructure and 5G networks, it would be beneficial to contrast its efficiency with other mechanisms. Moreover, four topologies—Tree, Abilene, GEANT, and DTelekom—were used to produce the total simulation findings. However, comparing the performance of CPVP to other indicators would also be helpful.

Enhancing the QoE of Customers: Today, to improve the QoE of the customers, the congestion control policy should cooperate with other QoS mechanisms, such as traffic conditioning and queue scheduling, so that the combined output can assist us in providing less E2EDelay and increasing the server load reduction, average cache hit ratio, and network footprint reduction.

## REFERENCES

- [1] D. Zhang, D. Zhou, and X. Jin, “Chunk mode VM migration in XIA and triple-way pipeline for performance optimization,” *Future Generation Computer Systems*, vol. 74, pp. 32–40, Sep. 2017.
- [2] J. Yang, F. Hu, X. Wang, D. Zhang, and Y. Wu, “Cross border Integration and Innovation of Intangible Cultural Heritage Elements in Contemporary Cultural and Creative Design,” in *2024 International Conference on Social Science, Education and Management (ICSSEM 2024)*. Francis: Academic Press, 2024.
- [3] H. Liu and R. Han, “A Hierarchical Cache Size Allocation Scheme Based on Content Dissemination in Information-Centric Networks,” *Future Internet*, vol. 13, no. 5, p. 131, 2021.
- [4] D. Man, Y. Wang, H. Wang, J. Guo, J. Lv, S. Xuan, and W. Yang, “Information-Centric Networking Cache Placement Method Based on Cache Node Status and Location,” *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, p. 5648765, 2021.
- [5] J. Cao, S. Chen, Z. Chen, G. Ma, Z. Yuan, Z. Hu, J. Zhou, and J. Guo, “Underlay implementation of Named Data Networking,” in *The Sixth International Conference on Networking and Distributed Computing*, 2016.
- [6] O. Serhane, K. Yahyaoui, B. Nour, and H. Mounsla, “A survey of ICN content naming and in-network caching in 5G and beyond networks,” *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4081–4104, 2020.
- [7] S. Choudhury, S. Mukherjee, P. Karimi, and D. Raychaudhuri, “nCore: Clean Slate Next-G Mobile Core Network Architecture for Scalability and Low Latency,” in *Network Security Empowered by Artificial Intelligence*. Springer, 2024, pp. 3–24.
- [8] T. A. Wibowo, N. Syambas, L. Hendrawan, and R. A. Yovita, “Closer Towards Named Data Networking Implementation,” *Int. J. Intell. Eng. Syst.*, vol. 16, no. 1, pp. 265–276, 2023.
- [9] P. Chaudhary, N. Hubballi, and S. G. Kulkarni, “eNCache: Improving content delivery with cooperative caching in Named Data Networking,” *Computer Networks*, vol. 237, p. 110104, 2023.
- [10] M. Andersson, “Enhancing content discovery in Video on Demand services for children,” Master Thesis, UMEA University, 2017.
- [11] N. Telesmanic and M. Elasmr, “Young Adults and Video Streaming Service Usage,” 2023.
- [12] Q. Fan, J. Li, X. Li, Q. He, S. Fu, and S. Wang, “Pa-cache: Learning-based popularity-aware content caching in edge networks,” *arXiv preprint arXiv:2002.08805*, 2020.

- [13] S. Kumar and R. Tiwari, "Optimized content centric networking for future Internet: dynamic popularity window based caching scheme," *Computer Networks*, vol. 179, p. 107434, 2020.
- [14] C. Y. Chang and M. S. Chang, "A hybrid coordination approach of in-network caching for named data networking," *International Journal of Future Generation Communication and Networking*, vol. 9, no. 4, pp. 285–300, 2016.
- [15] M. Wilbert, "The 15 Best VOD Platforms for Video On Demand in 2022 | Dacast," Jul. 2022. [Online]. Available: <https://www.dacast.com/blog/top-5-vod-platforms-online-video-hosting/>
- [16] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for Information-Centric Networks," in *Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking - ICN '12*. Helsinki: ACM, Aug. 2012, pp. 55–60.
- [17] L. Cui, E. Ni, Y. Zhou, Z. Wang, L. Zhang, J. Liu, and Y. Xu, "Towards real-time video caching at edge servers: A cost-aware deep Q-learning solution," *IEEE Transactions on Multimedia*, vol. 25, pp. 302–314, 2021.
- [18] X. Zhang, Y. Zhou, D. Wu, M. Hu, X. Zheng, M. Chen, and S. Guo, "Optimizing video caching at the edge: A hybrid multi-point process approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 10, pp. 2597–2611, 2022.
- [19] H. Asaeda, K. Matsuzono, Y. Hayamizu, H. H. Hlaing, and A. Ooka, "A survey of information-centric networking: The quest for innovation," *IEICE Transactions on Communications*, vol. 107, no. 1, pp. 139–153, 2024.
- [20] K. H. T. Chiu, J. M. Wang, A. M. Abdelmoniem, and B. Bensaou, "A two-tiered caching scheme for information-centric networks," in *2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR)*. IEEE, 2021, pp. 1–6.
- [21] P. Podder and A. Afanasyev, "On improving versatility of versec trust schema," in *Proceedings of the 9th ACM Conference on Information-Centric Networking*, 2022, pp. 174–176.
- [22] K. Spiteri, R. Uргаonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," *IEEE/ACM transactions on networking*, vol. 28, no. 4, pp. 1698–1711, 2020.
- [23] J. Shepherd, "30 Vital Video Marketing Statistics You Need to Know in 2024," Aug. 2024. [Online]. Available: <https://thesocialshepherd.com/blog/video-marketing-statistics>
- [24] J. Clement, "Video game market revenue worldwide from 2019 to 2029," Jun. 2024.
- [25] Q. Yang, H. Deng, and L. Wang, "A lightweight caching decision strategy based on node edge-degree for information centric networking," *IEEE Access*, 2020.

- [26] N. Dutta, S. K. Patel, O. S. Faragallah, M. Baz, and A. N. Z. Rashed, “Caching scheme for information-centric networks with balanced content distribution,” *International Journal of Communication Systems*, vol. 35, no. 7, p. e5104, 2022.
- [27] M. Gritter and D. R. Cheriton, “An architecture for content routing support in the Internet,” in *USITS*, vol. 1, 2001, pp. 4–16.
- [28] C. Yi, J. Abraham, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, “On the role of routing in Named Data Networking,” in *Proceedings of the 1st International conference on Information-centric networking - INC '14*. Paris: ACM, Sep. 2014, pp. 27–36.
- [29] M. Palmer, M. Appel, K. Spiteri, B. Chandrasekaran, A. Feldmann, and R. K. Sitaraman, “VOXEL: Cross-layer optimization for video streaming with imperfect transmission,” in *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, 2021, pp. 359–374.
- [30] X. Liu, F. Dobrian, H. Milner, J. Jiang, V. Sekar, I. Stoica, and H. Zhang, “A case for a coordinated internet video control plane,” in *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*. ACM, 2012, pp. 359–370.
- [31] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, “A survey of Information-Centric Networking research,” *IEEE Communications Surveys and Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.
- [32] I. U. Din, S. Hassan, M. K. Khan, M. Guizani, O. Ghazali, and A. Habbal, “Caching in Information-Centric Networking: Strategies, challenges, and future research directions,” *IEEE Communications Surveys and Tutorials*, vol. 20, no. 2, pp. 1443–1474, 2018.
- [33] D. Saxena, V. Raychoudhury, N. Suri, C. Becker, and J. Cao, “Named Data Networking: A survey,” *Computer Science Review*, vol. 19, pp. 15–55, feb 2016.
- [34] B. Nour, S. Mastorakis, R. Ullah, and N. Stergiou, “Information-centric networking in wireless environments: Security risks and challenges,” *IEEE wireless communications*, vol. 28, no. 2, pp. 121–127, 2021.
- [35] S. Wang, J. Bi, and J. Wu, “Cache policy performance for Information-Centric Networking under a hop-number-based metric framework,” *Journal of Internet Technology*, vol. 17, no. 3, pp. 409–420, 2016.
- [36] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, “A data-oriented (and beyond) network architecture,” *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 181–192, Aug. 2007.
- [37] W. Li, E. Chan, G. Feng, D. Chen, and S. Lu, “Analysis and performance study for coordinated hierarchical cache placement strategies,” *Computer Communications*, vol. 33, no. 15, pp. 1834–1842, sep 2010.

- [38] Q. Wu, W. Wang, P. Fan, Q. Fan, H. Zhu, and K. B. Letaief, “Cooperative edge caching based on elastic federated and multi-agent deep reinforcement learning in next-generation networks,” *IEEE Transactions on Network and Service Management*, 2024.
- [39] K. T. Sundararajan, V. Porpodas, T. M. Jones, N. P. Topham, and B. Franke, “Cooperative partitioning: Energy-efficient cache partitioning for high-performance CMPs,” in *IEEE International Symposium on High-Performance Comp Architecture*. IEEE, 2012, pp. 1–12.
- [40] Y. Chen, Y. Cai, H. Zheng, J. Hu, and J. Li, “Cooperative caching for scalable video coding using value-decomposed dimensional networks,” *China Communications*, vol. 19, no. 9, pp. 146–161, 2022.
- [41] W. Li, S. M. Oteafy, M. Fayed, and H. S. Hassanein, “Quality of experience in icn: Keep your low-bitrate close and high-bitrate closer,” *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 557–570, 2020.
- [42] W. Li, O. Sharief, M. Fayed, and H. S. Hassanein, “Bitrate adaptation-aware cache partitioning for video streaming over Information-centric Networks,” in *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*. Chicago, IL, USA: IEEE, 2018, pp. 401–408.
- [43] Y. Hayamizu, A. Ooka, K. Matsuzono, and H. Asaeda, “Controller-Assisted Adaptive Video Streaming Experimented in Cloud-Native ICN Platform,” in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2023, pp. 1–6.
- [44] M. Rezazad and Y. C. Tay, “A cache miss equation for partitioning an NDN content store,” in *Proceedings of the 9th Asian Internet Engineering Conference on - AINTEC '13*. Chiang Mai: ACM, nov 2013, pp. 1–8.
- [45] L. V. Yovita, N. R. Syambas, and I. J. M. Edward, “Weighted-CAPIC caching algorithm for priority traffic in named data network,” *Future Internet*, vol. 14, no. 3, p. 84, 2022.
- [46] M. T. Jahangir, A. Hussain, M. U. Farooq, A. A. Khan, and H. A. Bhutta, “Analyzing Cache Server Placement’s Impact On SDN-Based Cooperative Caching,” *Journal of Computing & Biomedical Informatics*, 2024.
- [47] L. Wang, S. Bayhan, and J. Kangasharju, “Optimal chunking and partial caching in information-centric networks,” *Computer Communications*, vol. 61, pp. 48–57, 2015.
- [48] R. M. Negara, N. P. Wasesa, Z. Muhammad, R. Mayasari, and S. Astuti, “Impact of Segmentation and Popularity-based Cache Replacement Policies on Named Data Networking,” *Jurnal Rekayasa Elektrika*, vol. 20, no. 1, 2024.
- [49] X. Liu, Z. Li, P. Yang, and Y. Dong, “Information-centric mobile Ad Hoc networks and content routing: A survey,” *Ad Hoc Networks*, vol. 58, pp. 255–268, 2017.

- [50] K. Amuda, W. Almustapha, B. Deepak, C. Hoggard, and P. Tiruveedula, "Revolutionizing Networking Paradigms: A Comprehensive Exploration of Information-Centric Networking (ICN), Content-Centric Networking (CCNx) and Named Data Networking (NDN)," *arXiv preprint arXiv:2407.02667*, 2024.
- [51] W. M. H. Azamuddin, A. H. Mohd Aman, H. Sallehuddin, M. Salam, and K. Abualsaud, "Mathematical Models for Named Data Networking Producer Mobility Techniques: A Review," *Mathematics*, vol. 12, no. 5, p. 649, 2024.
- [52] T. X. Tran and D. Pompili, "Octopus: A Cooperative Hierarchical Caching Strategy for Cloud Radio Access Networks," in *Proceedings - 2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2016*. Brasilia: IEEE, Oct. 2016, pp. 154–162.
- [53] I. Cisco, "Cisco Annual Internet Report (2018–2023) White Paper," Mar. 2020.
- [54] M. Falkner and J. Apostolopoulos, "Intent-based networking for the enterprise: a modern network architecture," *Communications of the ACM*, vol. 65, no. 11, pp. 108–117, 2022.
- [55] X. K. Zou, J. Erman, V. Gopalakrishnan, E. Halepovic, R. Jana, X. Jin, J. Rexford, and R. K. Sinha, "Can accurate predictions improve video streaming in cellular networks?" in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*. Santa Fe, New Mexico: ACM, Feb. 2015, pp. 57–62.
- [56] G. Dimopoulos, I. Leontiadis, P. Barlet-Ros, and K. Papagiannaki, "Measuring video QoE from encrypted traffic," in *Proceedings of the 2016 ACM on Internet Measurement Conference - IMC '16*. Santa Monica, California: ACM, Nov. 2016, pp. 513–526.
- [57] A. Galanopoulos, G. Iosifidis, A. Argyriou, and L. Tassiulas, "Green video delivery in LTE-based heterogeneous cellular networks," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a*. Boston, MA: IEEE, Jun. 2015, pp. 1–9.
- [58] H. Jin, D. Xu, C. Zhao, and D. Liang, "Information-Centric Mobile caching network frameworks and caching optimization: A survey," *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, p. 33, Dec. 2017.
- [59] R. Tourani, S. Misra, T. Mick, and G. Panwar, "Security, privacy, and access control in Information-Centric Networking: A survey," *Communications Surveys and Tutorials IEEE*, vol. 20, pp. 556–600, 2018.
- [60] C. Huang, J. Li, and K. W. Ross, "Can internet video-on-demand be profitable?" *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 133–144, 2007.
- [61] C. Ge, N. Wang, S. Skillman, G. Foster, and Y. Cao, "QoE-driven DASH video caching and adaptation at 5G mobile edge," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*. Kyoto: ACM, Sep. 2016, pp. 237–242.

- [62] W. A. Kamil, S. A. Nor, and R. Alubady, "Performance evaluation of TCP, UDP and DCCP traffic over 4G network," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 11, no. 10, pp. 1048–1057, 2015.
- [63] P. Zhang, A. Gember-Jacobson, Y. Zuo, Y. Huang, X. Liu, and H. Li, "Differential network analysis," in *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, 2022, pp. 601–615.
- [64] A. Lertsinsrubtavee, P. Mekbungwan, and N. Weshsuwannarugs, "Comparing NDN and CDN performance for content distribution service in community wireless Mesh network," in *Proceedings of the AINTEC 2014 on Asian Internet Engineering Conference*. Chiang Mai: ACM, Nov. 2014, pp. 43–50.
- [65] N. Aloulou, M. Ayari, M. F. Zhani, L. Saidane, and G. Pujolle, "Taxonomy and comparative study of NDN forwarding strategies," in *2017 sixth international conference on communications and networking (ComNet)*. Hammamet: IEEE, Mar. 2017, pp. 1–8.
- [66] C. Pu, "ProNDN: MCDM-Based Interest Forwarding and Cooperative Data Caching for Named Data Networking," *Journal of Computer Networks and Communications*, vol. 2021, no. 1, p. 6640511, 2021.
- [67] Y. Gui and Y. Chen, "A Cache Placement Strategy Based on Compound Popularity in Named Data Networking," *IEEE Access*, vol. 8, pp. 196 002–196 012, 2020.
- [68] A. Abrar, A. S. C. M. Arif, K. M. Zaini, M. H. Omar, and Y. Meng, "Advancing producer mobility management in Named Data Networking: A comprehensive analytical model," *Journal of King Saud University-Computer and Information Sciences*, vol. 36, no. 4, p. 102045, 2024.
- [69] K. Damayanti and A. Suyanto, "Positioning analysis of Video-on-Demand service provider in Indonesia based on E-Service quality dimensions," *Asian Journal of Research in Business and Management*, vol. 4, no. 2, pp. 98–110, 2022.
- [70] J. M. Garbes, D. C. Harnecillo, C. A. Yu, E. Pantoja *et al.*, "The effectiveness of facebook advertisements of subscription-based Video-On-Demand streaming services to filipino millennials," *Journal of Business and Management Studies*, vol. 4, no. 1, pp. 1–7, 2022.
- [71] L. Alkwai, A. Belghith, A. Gazdar, and S. AlAhmadi, "Transparent consumer mobility management in named data networking under the push communication mode," *Computer Networks*, vol. 235, p. 109953, 2023.
- [72] Y. Sun, S. K. Fayaz, Y. Guo, V. Sekar, Y. Jin, M. A. Kaafar, and S. Uhlig, "Trace-driven analysis of ICN caching algorithms on Video-on-Demand workloads," in *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*. Sydney: ACM, Dec. 2014, pp. 363–376.

- [73] I. Shemsi and P. Kadam, "Named data networking in vanet: A survey," *International Journal of Scientific Engineering and Science*, vol. 1, no. 11, pp. 45–49, 2017.
- [74] M. Ali Naeem, S. Awang Nor, S. Hassan, and B. S. Kim, "Compound popular content caching strategy in named data networking," *Electronics (Switzerland)*, vol. 8, no. 7, p. 771, 2019.
- [75] I. U. Din, S. Hassan, and A. Habbal, "A content placement scheme for Information-Centric Networking," *Advanced Science Letters*, vol. 21, no. 11, pp. 3482–3484, 2015.
- [76] H. A. Marhoon, M. Mahmuddin, S. A. Nor, M. H. Hussein, and A. S. Gifal, "Performance evaluation of CCM and TSCP routing protocols within/without data fusing in WSNs," *Journal of Physics: Conference Series*, vol. 1032, no. 1, pp. 1–12, 2018.
- [77] B. Nour, H. Khelifi, R. Hussain, S. Mastorakis, and H. Mougla, "Access control mechanisms in named data networks: A comprehensive survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–35, 2021.
- [78] J. Yang, Y. Yue, and K. V. Rashmi, "A large-scale analysis of hundreds of in-memory key-value cache clusters at twitter," *ACM Transactions on Storage (TOS)*, vol. 17, no. 3, pp. 1–35, 2021.
- [79] J. Zhao, T. Wen, H. Jahanshahi, and K. H. Cheong, "The random walk-based gravity model to identify influential nodes in complex networks," *Information Sciences*, vol. 609, pp. 1706–1720, 2022.
- [80] A. G. Sheshjavani, A. Khonsari, M. Moradian, S. P. Shariatpanahi, and S. B. Hassanpour, "Hybrid Coded-Uncoded Caching in Multi-Access Networks with Non-uniform Demands," *arXiv preprint arXiv:2401.07288*, 2024.
- [81] I. U. Din, S. Hassan, and A. Habbal, "Redundancy elimination in the future internet," in *Proceedings of the International Conference on Computing, Mathematics and Statistics (iCMS 2015) Bridging Research Endeavors*. Springer, 2017, pp. 67–73.
- [82] S.-H. Lim, Y. B. Ko, G. H. Jung, J. Kim, and M. W. Jang, "Inter-chunk popularity-based edge-first caching in Content-Centric Networking," *IEEE Communications Letters*, vol. 18, no. 8, pp. 1331–1334, 2014.
- [83] S. Kumar and R. Tiwari, "Dynamic popularity window and distance-based efficient caching for fast content delivery applications in CCN," *Engineering Science and Technology, an International Journal*, vol. 24, no. 3, pp. 829–837, 2021.
- [84] L. Gkatzikis, V. Sourlas, C. Fischione, and I. Koutsopoulos, "Low complexity content replication through clustering in Content-Delivery Networks," *Computer Networks*, vol. 121, pp. 137–151, 2017.

- [85] Z. Naor, S. K. Das, and M. Raj, "Content placement for Video-on-Demand services over cellular networks," *Wireless Personal Communications*, vol. 98, no. 1, pp. 467–486, 2018.
- [86] J. Kaur and S. Das, "Rspp: Restricted static pseudo-partitioning for mitigation of cross-core covert channel attacks," *ACM Transactions on Design Automation of Electronic Systems*, vol. 29, no. 2, pp. 1–22, 2024.
- [87] C. Xu, P. Liu, W. Wang, and Z. Li, "Evaluation of the impacts of traffic states on crash risks on freeways," *Accident Analysis & Prevention*, vol. 47, pp. 162–171, 2012.
- [88] L. JIANG, L. Bai, and Y.-m. WU, "Coupling and coordinating degrees of provincial economy, resources and environment in China," *Journal of natural resources*, vol. 32, no. 5, pp. 788–799, 2017.
- [89] G. Ausiello and L. Laura, "Directed hypergraphs: Introduction and fundamental algorithms—a survey," *Theoretical Computer Science*, vol. 658, pp. 293–306, 2017.
- [90] C. Koch, "Proactive Mechanisms For Video-on-demand Content Delivery," Doctoral thesis, Technischen Universität Darmstadt, 2018.
- [91] G. S. Paschos, G. Iosifidis, M. Tao, D. Towsley, and G. Caire, "The role of caching in future communication systems and networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1111–1125, jun 2018.
- [92] J. Ali, M. Adnan, T. R. Gadekallu, R. H. Jhaveri, and B.-H. Roh, "A QoS-aware software defined mobility architecture for named data networking," in *2022 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2022, pp. 444–449.
- [93] Q. Liu and Z. Fang, "Decentralized scheduling with qos constraints: Achieving o(1) qos regret of multi-player bandits," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 12, 2024, pp. 13 981–13 989.
- [94] Y.-D. Lin, "Second quarter 2018 IEEE communications surveys and tutorials," *IEEE Communications Surveys and Tutorials*, vol. 20, no. 2, pp. 829–835, 2018.
- [95] G. Huang, B. Zhang, Z. Yao, and C. Li, "Quality-aware video streaming for green cellular networks with hybrid energy sources," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8543–8556, 2020.
- [96] D. Niyato, "Second Quarter 2024 IEEE Communications Surveys and Tutorials," *IEEE Communications Surveys and Tutorials*, vol. 26, no. 2, pp. i–vi, 2024.
- [97] X. Tan, L. Xu, J. Ni, S. Li, X. Jiang, and Q. Zheng, "Game Theory Based Dynamic Adaptive Video Streaming for Multi-client over NDN," *IEEE Transactions on Multimedia*, vol. PP, 2021.
- [98] G. Parisis, V. Sourlas, K. V. Katsaros, W. K. Chai, G. Pavlou, and I. Wakeman, "Efficient content delivery through fountain coding in opportunistic information-centric networks," *Computer Communications*, vol. 100, pp. 118–128, 2017.

- [99] Y. Chen, J. Hu, J. Zhao, and G. Min, "Qos-aware computation offloading in leo satellite edge computing for iot: A game-theoretical approach," *Chinese Journal of Electronics*, vol. 33, no. 4, pp. 875–885, 2024.
- [100] W. Wang, Y. Sun, Y. Guo, D. Kaafar, J. Jin, J. Li, and Z. Li, "CRCache: Exploiting the correlation between content popularity and network topology information for ICN caching," in *2014 IEEE international conference on communications (ICC)*. Sydney, NSW: IEEE, Jun. 2014, pp. 3191–3196.
- [101] M. A. Naeem, R. Ullah, Y. Meng, R. Ali, and B. A. Lodhi, "Caching content on the network layer: A performance analysis of caching schemes in ICN-based Internet of Things," *IEEE Internet of Things Journal*, vol. 9, no. 9, pp. 6477–6495, 2021.
- [102] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. K. Ramakrishnan, "Optimal content placement for a Large-Scale VoD system," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2114–2127, 2016.
- [103] A. Azab, M. Khasawneh, S. Alrabaee, K.-K. R. Choo, and M. Sarsour, "Network traffic classification: Techniques, datasets, and challenges," *Digital Communications and Networks*, vol. 10, no. 3, pp. 676–692, 2024.
- [104] X. Wang, J. Lv, M. Huang, K. Li, J. Li, and K. Ren, "Energy-efficient ICN routing mechanism with QoS support," *Computer Networks*, vol. 131, pp. 38–51, 2018.
- [105] Q. N. Nguyen, R. Ullah, B.-S. Kim, R. Hassan, T. Sato, and T. Taleb, "A cross-layer green information-centric networking design toward the energy internet," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 3, pp. 1577–1593, 2022.
- [106] R. Alubady, M. Salman, and A. S. Mohamed, "A review of modern caching strategies in named data network: Overview, classification, and research directions," *Telecommunication Systems*, vol. 84, no. 4, pp. 581–626, 2023.
- [107] J. Khan, C. Westphal, and Y. Ghamri-doudane, "A popularity-aware centrality metric for content placement in Information Centric Networks," in *ICNC 2018-International Conference on Computing, Networking and Communication*. Maui, HI: IEEE, mar 2018, p. 8.
- [108] A. H. Raaid Alubady, Suhaidi Hassan, "Adaptive interest lifetime in Named Data Networking to support disaster area," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 2-4, pp. 29–34, 2018.
- [109] A. Afanasyev and S. K. Ramani, "NDNconf: Network management framework for named data networking," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2020, pp. 1–6.
- [110] F. A. Karim, A. H. M. Aman, R. Hassan, K. Nisar, and M. Uddin, "Named data networking: a survey on routing strategies," *IEEE Access*, vol. 10, pp. 90 254–90 270, 2022.

- [111] D. Kanellopoulos, “Academic video lectures over named data networking,” *Information Discovery and Delivery*, vol. 48, no. 4, pp. 165–177, 2020.
- [112] F. Farhan, D. Leman *et al.*, “Implementasi Metode Rivest Shamir Adleman (RSA) Untuk Kerahasiaan Database Perum Bulog Kanwil SUMUT,” *Journal of Machine Learning and Data Analytics*, vol. 2, no. 1, pp. 18–27, 2023.
- [113] V. K. Prasad, M. Shah, N. Patel, and M. Bhavsar, “Inspection of trust based cloud using security and capacity management at an IaaS level,” *Procedia Computer Science*, vol. 132, pp. 1280–1289, 2018.
- [114] M. W. Al Azad, R. Tourani, A. Mtibaa, and S. Mastorakis, “Harpocrates: Anonymous data publication in named data networking,” in *Proceedings of the 27th ACM on Symposium on Access Control Models and Technologies, 2022*, pp. 79–90.
- [115] A. Abrar, A. S. C. M. Arif, and K. M. Zaini, “Producer mobility support in information-centric networks: research background and open issues,” *International Journal of Communication Networks and Distributed Systems*, vol. 28, no. 3, pp. 312–336, 2022.
- [116] C. Fang, H. Yao, Z. Wang, W. Wu, X. Jin, and F. R. Yu, “A survey of mobile Information-Centric Networking: Research issues and challenges,” *IEEE Communications Surveys and Tutorials*, p. 20, 2018.
- [117] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. N. N. H. N. N. H. Briggs, R. L. Braynard, R. R. Ferreira, V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. N. N. H. N. N. H. Briggs, and R. L. Braynard, “Networking Named Content,” *Communications of the ACM*, vol. 55, no. 1, pp. 1–12, Dec. 2009.
- [118] V. Jacobson, J. Burke, L. Zhang, B. Zhang, K. Claffy, C. Papadopoulos, T. Abdelzaher, L. Wang, J. A. Halderman, and P. Crowley, “Named Data Networking next phase (NDN-NP) project May 2014-April 2015 annual report,” *no. April*, pp. 1–67, 2015.
- [119] W. M. H. Azamuddin, A. H. M. Aman, H. Sallehuddin, K. Abualsaud, and N. Mansor, “The emerging of named data networking: Architecture, application, and technology,” *IEEE Access*, vol. 11, pp. 23 620–23 633, 2023.
- [120] E. T. d. Silva, J. M. H. d. Macedo, and A. L. D. Costa, “NDN content store and caching policies: Performance evaluation,” *Computers*, vol. 11, no. 3, p. 37, 2022.
- [121] B. Alaya, L. Sellami, and M. Al Mutiq, “Towards a Dynamic Vehicular Clustering Improving VoD Services on Vehicular Ad Hoc Networks,” in *International Conference on Computational Collective Intelligence*. Springer, 2022, pp. 409–422.
- [122] N. Yang, K. Chen, and M. Wang, “SmartDetour: Defending blackhole and content poisoning attacks in IoT NDN networks,” *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 12 119–12 136, 2021.

- [123] S. Kalafatidis, S. Skaperas, V. Demiroglou, L. Mamatas, and V. Tsaoussidis, “Logically-centralized SDN-based NDN strategies for wireless mesh smart-city networks,” *Future Internet*, vol. 15, no. 1, p. 19, 2022.
- [124] W. M. H. Azamuddin, A. H. M. Aman, R. Hassan, and T.-A. N. Abdali, “Named data networking mobility: A survey,” in *International Conference on Emerging Technology Trends in Internet of Things and Computing*. Springer, 2021, pp. 266–281.
- [125] D. Gupta, S. Rani, S. H. Ahmed, and R. Hussain, “Caching policies in NDN-IoT architecture,” *Integration of WSN and IoT for Smart Cities*, pp. 43–64, 2020.
- [126] N. Yang, K. Chen, and Y. Liu, “Towards efficient NDN framework for connected vehicle applications,” *IEEE Access*, vol. 8, pp. 60 850–60 866, 2020.
- [127] T. Tuyen, N. D. N. Ha, N. Loc *et al.*, “Exploring The Relationship Between ChatGPT and Task-based Approach: Opportunities and Challenges,” in *20th International Conference of the Asia Association of Computer-Assisted Language Learning (AsiaCALL 2023)*. Atlantis Press, 2024, pp. 22–36.
- [128] I. Dasgupta, S. Shannigrahi, and M. Zink, “A hybrid NDN-IP architecture for live video streaming: From host-based to content-based delivery to improve QoE,” *International Journal of Semantic Computing*, vol. 16, no. 02, pp. 163–187, 2022.
- [129] S. Mishra, V. K. Jain, K. Gyoda, and S. Jain, “Distance-based dynamic caching and replacement strategy in NDN-IoT networks,” *Internet of Things*, vol. 27, p. 101264, 2024.
- [130] F. Zharfan, L. D. Hasnaa, R. M. Negara, and N. R. Syambas, “Comparison of Caching Replacement Policies in Changing the Number of Interest Packets on Named Data Networks Using Mininet-NDN,” in *2021 15th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*. IEEE, 2021, pp. 1–8.
- [131] S. J. Taher, O. Ghazali, and S. Hassan, “A review on cache replacement strategies in Named Data Network,” *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 2-4, pp. 53–57, 2018.
- [132] H. L. Hasan and S. A. Al-Bermany, “A weak relation-based approach to detect influential nodes in complex network,” in *AIP Conference Proceedings*, vol. 2398, no. 1. AIP Publishing, 2022.
- [133] N. Bendahman and D. Lotfi, “Unveiling Influence in Networks: A Novel Centrality Metric and Comparative Analysis through Graph-Based Models,” *Entropy*, vol. 26, no. 6, p. 486, 2024.
- [134] C. Guo, L. Yang, X. Chen, D. Chen, H. Gao, and J. Ma, “Influential nodes identification in complex networks via information entropy,” *Entropy*, vol. 22, no. 2, p. 242, 2020.

- [135] R. Zhang, J. Liu, R. Xie, T. Huang, F. R. Yu, and Y. Liu, "Service-aware optimal caching placement for named data networking," *Computer Networks*, vol. 174, p. 107193, 2020.
- [136] J. Ran, N. Lv, D. Zhang, Y. Ma, and Z. Xie, "On performance of cache policies in Named Data Networking," in *International Conference on Advanced Computer Science and Electronics Information*, no. Icacsei. Atlantis, 2013, pp. 668–671.
- [137] E. Ghabashneh and S. Rao, "Exploring the interplay between CDN caching and video streaming performance," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 516–525.
- [138] Y. Gui and Y. Chen, "A cache placement strategy based on entropy weighting method and TOPSIS in named data networking," *IEEE Access*, vol. 9, pp. 56 240–56 252, 2021.
- [139] D. Doan Van and Q. Ai, "In-network caching in information-centric networks for different applications: A survey," *Cogent Engineering*, vol. 10, no. 1, p. 2210000, 2023.
- [140] S. Wang, J. Bi, J. Wu, and A. V. Vasilakos, "CPHR: In-network caching for information-centric networking with partitioning and hash-routing," *IEEE/ACM transactions on networking*, vol. 24, no. 5, pp. 2742–2755, 2015.
- [141] H. Noh and H. Song, "Progressive caching system for video streaming services over content centric network," *IEEE Access*, vol. 7, pp. 47 079–47 089, 2019.
- [142] S. Lederer, C. Mueller, C. Timmerer, and H. Hellwagner, "Adaptive multimedia streaming in Information-Centric Networks," *IEEE Network*, no. 6, pp. 91–96, 2014.
- [143] R. Gopalakrishnan, D. Kanoulas, N. N. Karuturi, C. P. Rangan, R. Rajaraman, and R. Sundaram, "Cache Me if You Can: Capacitated Selfish Replication Games in Networks," *Theory of Computing Systems*, vol. 64, pp. 272–310, 2020.
- [144] Y. Zeng and X. Hong, "A caching strategy in mobile Ad Hoc Named Data Network," in *2011 6th International ICST Conference on Communications and Networking in China (CHINACOM)*. Harbin: IEEE, Aug. 2011, pp. 805–809.
- [145] S. Taterh, Y. Meena, and G. Paliwal, "Performance analysis of ad hoc on-demand distance vector routing protocol for mobile ad hoc networks," *Computational Network Application Tools for Performance Management*, pp. 235–245, 2020.
- [146] H.-g. Choi, J. Yoo, T. Chung, N. Choi, T. Kwon, and Y. Choi, "CoRC: coordinated routing and caching for named data networking," in *Proceedings of the tenth ACM/IEEE symposium on Architectures for networking and communications systems*. ACM, 2014, pp. 161–172.
- [147] R. S. Abbas, S. A. Nor, and A. Ahmad, "A review of cache placement algorithms for Video on Demand in Named Data Networking," *Journal of Advanced*

*Research in Dynamical and Control Systems (JARDCS)*, vol. 10, no. 10-Special Issue, pp. 1322–1332, 2018.

- [148] H. Zhang, H. Liang, Y. Wang, D. Li, Z. Shao, M. Yi, Y. Lu, and Z. Huang, “Multi-transition delay test for improving the coverage of cell internal defects,” *IEICE Electronics Express*, vol. 21, no. 15, pp. 20 240 326–20 240 326, 2024.
- [149] D. Man, Q. Lu, H. Wang, J. Guo, W. Yang, and J. Lv, “On-path caching based on content relevance in information-centric networking,” *Computer Communications*, vol. 176, pp. 272–281, 2021.
- [150] S. Wang and Z. Ning, “Collaborative caching strategy in content-centric networking,” in *Advances in Computing, Informatics, Networking and Cybersecurity: A Book Honoring Professor Mohammad S. Obaidat’s Significant Scientific Contributions*. Springer, 2022, pp. 465–511.
- [151] C. Chen, C. Wang, T. Qiu, M. Atiquzzaman, and D. O. Wu, “Caching in vehicular named data networking: Architecture, schemes and future directions,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2378–2407, 2020.
- [152] L. V. Yovita, T. A. Wibowo, A. A. Ramadha, G. P. Satriawan, and S. Raniprma, “Performance Analysis of Cache Replacement Algorithm using Virtual Named Data Network Nodes,” *Jurnal Online Informatika*, vol. 7, no. 2, pp. 203–210, 2022.
- [153] B. P. Julian, K. Pahuja, and M. S. Sidhu, “Enhancements to content caching using weighted greedy caching algorithm in information centric networking,” *Procedia Computer Science*, vol. 171, pp. 2435–2444, 2020.
- [154] W. Huang, T. Song, Y. Yang, and Y. Zhang, “Cluster-based cooperative caching with mobility prediction in vehicular named data networking,” *IEEE Access*, vol. 7, pp. 23 442–23 458, 2019.
- [155] H. Che, Y. Tung, and Z. Wang, “Hierarchical web caching systems: Modeling, design and experimental results,” *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 7, pp. 1305–1314, Sep. 2002.
- [156] L. Saino, I. Psaras, and G. Pavlou, “Icarus: A caching simulator for Information Centric Networking (ICN),” in *Proceedings of the 7th International ICST Conference on Simulation Tools and Techniques*. Lisbon: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Mar. 2014, pp. 66–75.
- [157] M. Aminian, M. J. Shahbazzadeh, and M. Eslami, “Distance optimization and directional overcurrent relay coordination using edge-powered biogeography-genetic algorithms,” *Journal of Cloud Computing*, vol. 13, no. 1, p. 109, 2024.
- [158] P. Benjamin and A. Abate, “Networked Communication for Mean-Field Games with Function Approximation and Empirical Mean-Field Estimation,” *arXiv preprint arXiv:2408.11607*, 2024.

- [159] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache 'less for more' in Information-Centric Networks (extended version)," *Computer Communications*, vol. 36, no. 7, pp. 758–770, 2013.
- [160] L. Zeno, A. Chen, and M. Silberstein, "In-Network Address Caching for Virtual Networks," in *Proceedings of the ACM SIGCOMM 2024 Conference*, 2024, pp. 735–749.
- [161] C. Fan, S. Shannigrahi, C. Papadopoulos, and C. Partridge, "Discovering in-network caching policies in ndn networks from a measurement perspective," in *Proceedings of the 7th ACM Conference on Information-Centric Networking*, 2020, pp. 106–116.
- [162] J. Pfender, A. Valera, and W. K. G. Seah, "Content delivery latency of caching strategies for information-centric IoT," *arXiv preprint arXiv:1905.01011*, 2019.
- [163] N. Laoutaris, H. Che, and I. Stavrakakis, "The LCD interconnection of LRU caches and its analysis," *Performance Evaluation*, vol. 36, no. 7, pp. 609–63, 2006.
- [164] S. Eum, K. Nakauchi, M. Murata, Y. Shoji, and N. Nishinaga, "CATT: Potential based routing with content caching for ICN," in *Proceedings of the second edition of the ICN workshop on Information-centric networking*. Helsinki: ACM, aug 2012, pp. 49–54.
- [165] H. Wu, Y. Fan, Y. Wang, H. Ma, and L. Xing, "A comprehensive review on edge caching from the perspective of total process: Placement, policy and delivery," *Sensors*, vol. 21, no. 15, p. 5033, 2021.
- [166] J. Shuja, K. Bilal, W. Alasmary, H. Sinky, and E. Alanazi, "Applying machine learning techniques for caching in next-generation edge networks: A comprehensive survey," *Journal of Network and Computer Applications*, vol. 181, p. 103005, 2021.
- [167] A. Aboodi, T.-C. Wan, and G.-C. Sodhy, "Survey on the Incorporation of NDN/CCN in IoT," *IEEE Access*, vol. 7, pp. 71 827–71 858, 2019.
- [168] B. Makhkamov and N. Khasanov, "A survey of cache placement algorithms in content delivery networks," in *E3S Web of Conferences*, vol. 458. EDP Sciences, 2023, p. 09004.
- [169] M. Yamamoto, "A survey of caching networks in content oriented networks," *IEICE Transactions on Communications*, vol. 99, no. 5, pp. 961–973, 2016.
- [170] J. Li, H. Wu, B. Liu, and J. Lu, "Effective caching schemes for minimizing inter-ISP traffic in Named Data Networking," in *2012 IEEE 18th International Conference on Parallel and Distributed Systems*. Singapore: IEEE, Dec. 2012, pp. 580–587.
- [171] S. Bayhan, L. Wang, J. Ott, J. Kangasharju, A. Sathiaseelan, and J. Crowcroft, "On content indexing for off-path caching in Information-Centric Networks," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*. ACM, 2016, pp. 102–111.

- [172] M. M. F. Hamdi, A. Habbal, N. H. Zakaria, and S. Hassan, "Evaluation of caching strategies in content-centric networking for mobile and social networking environment," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 2-4, pp. 1–6, 2018. [Online]. Available: [journal.utem.edu.my/index.php/jtec/article/download/4298/3155](http://journal.utem.edu.my/index.php/jtec/article/download/4298/3155)
- [173] N. Laoutaris, S. Syntila, and I. Stavrakakis, "Meta algorithms for hierarchical Web caches," in *IEEE International Conference on Performance, Computing, and Communications, 2004*. Phoenix, AZ, USA: IEEE, Apr. 2004, pp. 445–452.
- [174] S. Hassan, W. Elbreiki, M. Firdhous, and A. M. Monzer Habbala, "End-to-End networks vs Named Data Network: A critical evaluation," *Jurnal Teknologi*, vol. 72, no. 5, pp. 71–76, 2015. [Online]. Available: [https://www.researchgate.net/profile/Mohamed\\_Firdhous/publication/276892372\\_End-to-End\\_Networks\\_Vs\\_Named\\_Data\\_Network\\_A\\_Critical\\_Evaluation/links/55bb2ec008aec0e5f43eb2f3.pdf](https://www.researchgate.net/profile/Mohamed_Firdhous/publication/276892372_End-to-End_Networks_Vs_Named_Data_Network_A_Critical_Evaluation/links/55bb2ec008aec0e5f43eb2f3.pdf)
- [175] B. Jedari, G. Premsankar, G. Illahi, M. Di Francesco, A. Mehrabi, and A. Ylä-Jääski, "Video caching, analytics, and delivery at the wireless edge: A survey and future directions," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 431–471, 2020.
- [176] R. Wu and W.-K. Jia, "An RNS-Based NDN Forwarding Plane: Design and Evaluation," *IEEE Access*, vol. 11, pp. 20 952–20 962, 2023.
- [177] X. Yin, V. Sekar, and B. Sinopoli, "Toward a principled framework to design dynamic adaptive streaming algorithms over HTTP," in *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*. Los Angeles, CA: ACM, Oct. 2014, p. 9.
- [178] J. Wang, J. Ren, K. Lu, J. Wang, S. Liu, and C. Westphal, "An optimal Cache management framework for information-centric networks with network coding," in *2014 IFIP Networking Conference*, no. 61202378. Trondheim, Norway: IEEE, Jun. 2014, pp. 1–9. [Online]. Available: <http://ieeexplore.ieee.org/document/6857127/>
- [179] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A survey of green Information-Centric Networking: Research issues and challenges," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 3, pp. 1455–1472, 2015. [Online]. Available: <https://ieeexplore.ieee.org/document/7035028/?denied>
- [180] A. T. Herouala, B. Ziani, C. A. Kerrache, A. e. Karim Tahari, N. Lagraa, and S. Mastorakis, "CaDaCa: a new caching strategy in NDN using data categorization," *Multimedia Systems*, vol. 29, no. 5, pp. 2935–2950, 2023.
- [181] L. Ramaswamy and L. Liu, "An expiration age-based document placement scheme for cooperative Web caching," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 5, pp. 585–600, May 2004. [Online]. Available: <http://ieeexplore.ieee.org/document/1277819/>

- [182] I. G. A. Zurgani, M. K. Elbashir, A. M. Ahmed, and T. A. O. Hassan, "A comparative study of in-network caching strategies for information centric networks," in *2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*. IEEE, 2019, pp. 1–6.
- [183] S. Saha, A. Lukyanenko, and Antti, "Efficient cache availability management in Information-Centric Networks," *Computer Networks*, vol. 84, pp. 32–45, Feb. 2015.
- [184] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang, "Interest flooding attack and countermeasures in Named Data Networking," in *IFIP Networking Conference, 2013*. Brooklyn, NY: IEEE, May 2013, pp. 1–9. [Online]. Available: <http://lasr.cs.ucla.edu/afanasyev/data/files/Afanasyev/ifip-interest-flooding-ndn.pdf>
- [185] Liangzhong Yin and Guohong Cao, "Supporting cooperative caching in Ad Hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 1, pp. 77–89, 2006. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1542018>
- [186] A. Compagno, M. Conti, C. Ghali, and G. Tsudik, "To NACK or not to NACK? Negative acknowledgments in Information-Centric Networking," in *Computer Communication and Networks (ICCCN), 2015 24th International Conference on*. Las Vegas, NV: IEEE, Aug. 2015, pp. 1–10. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/7288477/>
- [187] Q. Gao, J. Zhang, Y. Yin, G. Zhang, and X. Qian, "FoV-Aware Cache Coordination Schemes for SVC-Encoding 360-Degree Panoramic Streaming in ICN," in *2024 9th International Conference on Signal and Image Processing (ICSIP)*. IEEE, 2024, pp. 731–735.
- [188] J. Li, H. Wu, B. Liu, J. Lu, Y. Wang, X. Wang, Y. Zhang, and L. Dong, "Popularity-driven coordinated caching in named data networking," in *Proceedings of the eighth ACM/IEEE symposium on Architectures for networking and communications systems*, 2012, pp. 15–26.
- [189] S. Han, H. Su, C. Yang, and A. F. Molisch, "Proactive edge caching for video on demand with quality adaptation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 218–234, 2019.
- [190] Z. Zhang, H. Ma, Y. Xue, and L. Liu, "Fair video caching for named data networking," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.
- [191] R. Paul and M. P. Selvan, "A study on naming and caching in named data networking," in *2021 fifth international conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC)*. IEEE, 2021, pp. 1387–1395.
- [192] N. Walliman, *Research methods: The basics*, Taylor and Francis, Eds. London: Routledge, 2021.
- [193] N. K. Malhotra, D. Nunan, and D. F. Birks, *Marketing research*, F. Edition, Ed. Pearson UK, 2020.



- [207] S. Hassan, "Simulation-based performance evaluation of TCP-Friendly protocols for supporting multimedia application in the Internet," Doctoral Thesis, University of Leeds (School of Computing), 2002. [Online]. Available: <http://myto.upm.edu.my/myTO/myto/1/paparthesis/519648.html>
- [208] Y. Ren, J. Li, L. Li, S. Shi, J. Zhi, and H. Wu, "Modeling content transfer performance in Information-Centric Networking," *Future Generation Computer Systems*, vol. 74, pp. 12–19, Sep. 2017. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0167739X17305861>
- [209] J. Mo, *Performance modeling of communication networks with Markov chains*. Springer Nature, 2022.
- [210] C. K. Filelis-Papadopoulos, P. T. Endo, M. Bendeche, S. Svorobej, K. M. Giannoutakis, G. A. Gravvanis, D. Tzovaras, J. Byrne, and T. Lynn, "Towards simulation and optimization of cache placement on large virtual content distribution networks," *Journal of Computational Science*, vol. 39, p. 101052, 2020.
- [211] H. Sun, H. V. Burton, and H. Huang, "Machine learning applications for building structural design and performance assessment: State-of-the-art review," *Journal of Building Engineering*, vol. 33, p. 101816, 2021.
- [212] R. M. Fujimoto, K. S. Perumalla, and G. F. Riley, *Network simulation*. Springer Nature, 2022.
- [213] M. Hussaini, H. Awwalu, H. Bello-Salau, and J. Murtala, "Simulation Investigation for Named Data Networking Mobility Support," *Int. J. Inf. Process. Commun.*, vol. 9, pp. 258–271, 2020.
- [214] I. Yildirim, A. Uyrus, and E. Basar, "Modeling and analysis of reconfigurable intelligent surfaces for indoor and outdoor applications in future wireless networks," *IEEE transactions on communications*, vol. 69, no. 2, pp. 1290–1301, 2020.
- [215] H. Wu, J. U. N. Li, and L. Li, "Design and evaluation of probabilistic caching in Information-Centric Networking," *IEEE Access*, vol. 6, pp. 32 754–32 768, 2018. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8372903>
- [216] O. Dakkak, S. Arif, and S. Awang Nor, "A critical analysis of simulators in grid," *Jurnal Teknologi*, vol. 77, no. 4, pp. 111–117, 2015.
- [217] W. Du, F. K. Tuffner, K. P. Schneider, R. H. Lasseter, J. Xie, Z. Chen, and B. Bhattarai, "Modeling of grid-forming and grid-following inverters for dynamic simulation of large-scale distribution systems," *IEEE Transactions on Power Delivery*, vol. 36, no. 4, pp. 2035–2045, 2020.
- [218] L. Campanile, M. Gribaudo, M. Iacono, F. Marulli, and M. Mastroianni, "Computer network simulation with ns-3: A systematic literature review," *Electronics*, vol. 9, no. 2, p. 272, 2020.

- [219] S. Mastorakis, A. Afanasyev, and L. Zhang, “Public Review for On the Evolution of ndnSIM,” *ACM SIGCOMM Computer Communication Review*, vol. 47, no. 3, pp. 19–33, 2017.
- [220] E. Bardhi, M. Conti, R. Lazzeretti, E. Losiouk, and A. Taffal, “Sim2Testbed Transfer: NDN Performance Evaluation,” in *Proceedings of the 17th International Conference on Availability, Reliability and Security*, 2022, pp. 1–9.
- [221] S. U. Taki and S. Mastorakis, “An NDN-Enabled Fog Radio Access Network Architecture With Distributed In-Network Caching,” *arXiv preprint arXiv:2301.08564*, 2023.
- [222] N. Team and Others, “NDN packet format specification 0.2-2 documentation,” 2017. [Online]. Available: <http://named-data.net/doc/NDN-packet-spec/current/>
- [223] K. Delvadia, N. Dutta, and R. Jadeja, “CCJRF-ICN: A novel mechanism for coadjutant caching joint request forwarding in information centric networks,” *IEEE Access*, vol. 9, pp. 84 134–84 155, 2021.
- [224] V. Jacobson, “A new way to look at networking,” *Google Tech Talk*, vol. 30, 2006. [Online]. Available: <https://github.com/dominictarr/cyphernet/issues/15>
- [225] G. Jaber and R. Kacimi, “A collaborative caching strategy for content-centric enabled wireless sensor networks,” *Computer Communications*, vol. 159, pp. 60–70, 2020.
- [226] Y. Wang, J. Yang, and Z. Wang, “Dynamically configuring LRU replacement policy in Redis,” in *Proceedings of the International Symposium on Memory Systems*, 2020, pp. 272–280.
- [227] P. Moll, V. Patil, L. Wang, and L. Zhang, “SoK: The evolution of distributed dataset synchronization solutions in NDN,” in *Proceedings of the 9th ACM Conference on Information-Centric Networking*, 2022, pp. 33–44.
- [228] T. Ezaki, N. Imura, and K. Nishinari, “Towards understanding network topology and robustness of logistics systems,” *Communications in Transportation Research*, vol. 2, p. 100064, 2022.
- [229] T. J. Grant, *Network topology in command and control: Organization, operation, and evolution*. IGI Global, 2014.
- [230] Y. Li, T. Lin, H. Tang, and P. Sun, “A chunk caching location and searching scheme in Content Centric Networking,” in *In 2012 IEEE International Conference on Communications (ICC)*. Ottawa, ON: IEEE, Jun. 2012, pp. 2655–2659. [Online]. Available: <https://arxiv.org/pdf/1701.02524.pdf>
- [231] D. Rossi and G. Rossini, “Caching performance of content centric networks under multi-path routing (and more),” *Relatório técnico, Telecom ParisTech*, pp. 1–6, 2011. [Online]. Available: <http://netlab.pkusz.edu.cn/wordpress/wp-content/uploads/2011/10/>

Caching-performance-of-content-centric-networks-under-multi-path-routingand-more.pdf

- [232] K. Cho, M. Lee, K. Park, T. T. Kwon, Y. Choi, and S. Pack, "WAVE: Popularity-based and collaborative in-network caching for content-oriented networks," in *2012 Proceedings IEEE INFOCOM Workshops*. Orlando, FL: IEEE, Mar. 2012, pp. 316–321. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.456.2292&rep=rep1&type=pdf>
- [233] C. Bernardini, T. Silverston, and F. Olivier, "MPC : Popularity-based Caching Strategy for Content Centric Networks," in *2013 IEEE International Conference on Communications (ICC)*. Budapest: IEEE, Jun. 2014, pp. 3619–3623. [Online]. Available: <https://hal.inria.fr/file/index/docid/929737/filename/main.pdf>
- [234] I. Psaras, R. G. Clegg, R. Landa, W. K. Chai, and G. Pavlou, "Modelling and evaluation of CCN-caching trees," in *International Conference on Research in Networking*. Springer, 2011, pp. 78–91.
- [235] A. Suvorov, J. Hochuli, and D. R. Schrider, "Accurate inference of tree topologies from multiple sequence alignments using deep learning," *Systematic biology*, vol. 69, no. 2, pp. 221–233, 2020.
- [236] S. Yang, X. Qiu, H. Xie, J. Guan, Y. Liu, and C. Xu, "GDSoc: green dynamic self-optimizing content caching in ICN-based 5G network," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 1, p. e3221, 2018.
- [237] GÉANT, "GÉANT topology." [Online]. Available: [https://www.geant.org/Networks/Pan-European\\_network/Pages/GEANT\\_topology\\_map.aspx](https://www.geant.org/Networks/Pan-European_network/Pages/GEANT_topology_map.aspx)
- [238] D. Telekom, "Deutsche Telekom: Home." [Online]. Available: <https://www.telekom.com/en/company/topic-specials/digital-x-special>
- [239] R. K. Raj and A. N. Kumar, "Toward computer science curricular guidelines 2023 (cs2023)," *ACM Inroads*, vol. 13, no. 4, pp. 22–25, 2022.
- [240] J. Tang, Z. Zhang, Y. Liu, and H. Zhang, "Identifying interest flooding in Named Data Networking," in *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCoM), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*. Beijing: IEEE, Aug. 2013, pp. 306–310.
- [241] D. Posch, B. Rainer, S. Theuermann, A. Leibetseder, and H. Hellwagner, "Emulating NDN-based multimedia delivery," in *Proceedings of the 7th International Conference on Multimedia Systems - MMSys '16*. Klagenfurt: ACM, May 2016, p. 26.
- [242] H. Yang, H. Pan, and L. Ma, "A review on software defined content delivery network: A novel combination of CDN and SDN," *IEEE Access*, vol. 11, pp. 43 822–43 843, 2023.

- [243] M. Daffa, L. V. Yovita, and T. A. Wibowo, "The NDNs performance with variation of topology and prefix on named data networking," *[CEPAT] Journal of Computer Engineering: Progress, Application and Technology*, vol. 1, no. 03, pp. 47–57, 2022.
- [244] F. V. Mutlu and E. Yeh, "Cost-aware joint caching and forwarding in networks with heterogeneous cache resources," in *ICC 2024-IEEE International Conference on Communications*. IEEE, 2024, pp. 4736–4742.
- [245] M. A. Naeem, S. A. Nor, and S. Hassan, "EdgeTouch Caching Strategy for Video on Demand Data Contents in Named Data Networking," *Journal of Engineering Research and Education*, vol. 12, pp. 1–12, 2020.
- [246] R. N. B. Rais and O. Khalid, "Study and analysis of mobility, security, and caching issues in CCN," *International Journal of Electrical and Computer Engineering (2088-8708)*, vol. 10, no. 2, 2020.
- [247] M. A. Naeem, M. A. U. Rehman, R. Ullah, and B.-S. Kim, "A comparative performance analysis of popularity-based caching strategies in named data networking," *IEEE access*, vol. 8, pp. 50 057–50 077, 2020.
- [248] "Géant network website," Sep. 2024. [Online]. Available: <http://geant3.archive.geant.net/Network/>
- [249] T. Citrix and M. Analytics, "Citrix mobile analytics report September 2014: Mobile subscriber data usage trends," Sep. 2014.
- [250] M. A. Talaat, M. A. Koutb, and H. S. Sorour, "PSNR evaluation of media traffic over TFRC," *International Journal of Computer Networks and Communications*, vol. 1, no. 3, pp. 71–76, 2009.
- [251] BörseFrankfurt, "Börse frankfurt (Frankfurt Stock Exchange): Stock market quotes, charts and news," 2018. [Online]. Available: [http://en.boerse-frankfurt.de/index/constituents/Euro\\_Stoxx\\_50#Constituents](http://en.boerse-frankfurt.de/index/constituents/Euro_Stoxx_50#Constituents)
- [252] D. Telekom, "Deutsche Telekom: Inspired for education." [Online]. Available: <https://www.telekom.com/en/corporate-responsibility/corporate-responsibility/inspired-for-education-355392>
- [253] K. A. Raza, A. Asheralieva, M. M. Karim, K. Sharif, M. Gheisari, and S. Khan, "A novel forwarding and caching scheme for information-centric software-defined networks," in *2021 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 2021, pp. 1–8.
- [254] J. Zhang and E. Yeh, "LOAM: Low-latency Communication, Caching, and Computation Placement in Data-Intensive Computing Networks," *arXiv preprint arXiv:2403.15927*, 2024.
- [255] M. Hassan, H. Salat, M. Conti, F. H. P. Fitzek, and T. Strule, "CoMon-DAS: A framework for efficient and robust dynamic adaptive streaming over NDN," in *2019 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2019, pp. 1–7.

- [256] T. Leimbach, D. Hallinan, D. Bachlechner, A. Weber, M. Jaglo, L. Hennen, R. Ø. Nielsen, M. Nentwich, S. Strauß, T. Lynn, and G. Hunt, “Potential and Impacts of Cloud Computing Services and Social Network Websites,” *European Parliamentary Research Service*, no. January, pp. 1–142, 2014. [Online]. Available: <https://policycommons.net/artifacts/1339599/potential-and-impacts-of-cloud-computing-services-and-social-network-websites/1949429/>
- [257] C. Bernardini, “Stratégies de cache basées sur la popularité pour Content Centric Networking,” Doctoral Dissertation, Université de Lorraine, 2015. [Online]. Available: <https://www.theses.fr/2015LORR0121>
- [258] L. A. Adamic and B. A. Huberman, “Zipf’s law and the Internet,” *Glottometrics*, vol. 3, no. 1, pp. 143–150, 2002. [Online]. Available: <https://www.arteauna.com/talleres/lab/ediciones/libreria/Glottometrics-zipf.pdf#page=148>
- [259] L. Saino, I. Psaras, and G. Pavlou, “Hash-routing schemes for information centric networking.” in *ICN*. China: ACM, Aug. 2013, pp. 27–32.
- [260] M. Bilal and S.-G. Kang, “Time Aware Least Recent Used (TLRU) cache management policy in ICN,” in *16th International Conference on Advanced Communication Technology*. Pyeongchang, Korea (South): IEEE, Feb. 2014, pp. 528–532. [Online]. Available: <https://arxiv.org/ftp/arxiv/papers/1801/1801.00390.pdf>
- [261] D. O. Mau, M. Chen, T. Taleb, X. Wang, and V. C. M. Leung, “FGPC: Fine-Grained Popularity-based Caching design for Content Centric Networking,” in *MSWiM 2014 - Proceedings of the 17th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. New York, New York, USA: ACM Press, 2014, pp. 295–302. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2641798.2641837>
- [262] C. Bernardini, T. Silverston, and O. Festor, “Socially-aware caching strategy for content centric networking,” in *2014 IFIP Networking Conference*. Trondheim, Norway: IEEE, Jun. 2014, pp. 1–9. [Online]. Available: <http://ieeexplore.ieee.org/document/6857093/>
- [263] J. M. Wang, J. Zhang, and B. Bensaou, “Intra-AS cooperative caching for content-centric networks,” in *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-Centric Networking*, no. August. Hong Kong: ACM, aug 2013, pp. 61–66.
- [264] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. M. Maggs, K. C. Ng, V. Sekar, and S. Shenker, “Less pain, most of the gain: incrementally deployable ICN,” *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 147–158, Aug. 2013. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2486001.2486023>
- [265] E. J. Renssweig and J. Kurose, “Breadcrumbs: Efficient, best-effort content location in cache networks,” in *IEEE INFOCOM 2009 - The 28th Conference on Computer Communications*. Rio de Janeiro: IEEE, Apr. 2009, pp.

2631–2635. [Online]. Available: <https://web.cs.umass.edu/publication/docs/2009/UM-CS-2009-005.pdf>

- [266] J. Sung, J.-K. K. Rhee, and S. Jung, “Lightweight caching strategy for wireless content delivery networks,” *IEICE Communications Express*, vol. 3, no. 4, pp. 150–155, 2014.
- [267] G. Zhang, B. Tang, X. Wang, and Y. Wu, “An Optimal Cache Placement Strategy Based on Content Popularity in Content Centric Network,” *Journal of Information and Computational Science*, vol. 11, no. 8, pp. 2759–2769, 2014.
- [268] Z. Ming, M. Xu, and D. Wang, “Age-based cooperative caching in Information-Centric Networks,” in *Computer Communication and Networks (ICCCN), 2014 23rd International Conference on*. Shanghai: IEEE, aug 2014, pp. 268–273.
- [269] J. Ren, W. Qi, C. Westphal, J. Wang, K. Lu, S. Liu, and S. Wang, “MAGIC: A distributed MAX-Gain In-network Caching strategy in information-centric networks,” in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. Toronto, ON, Canada: IEEE, apr 2014, pp. 470–475. [Online]. Available: <http://ieeexplore.ieee.org/document/6849277/>
- [270] C. Kreuzberger, D. Posch, and H. Hellwagner, “A scalable video coding dataset and toolchain for dynamic adaptive streaming over HTTP,” in *Proceedings of the 6th ACM Multimedia Systems Conference*. Portland: ACM, Mar. 2015, pp. 213–218.
- [271] W. Liu, G. Zhang, and Q. Gao, “SVC-aware cache coordination schemes for dynamic adaptive streaming in named data networking,” in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.
- [272] M. R. Rotinsulu, B. Susilo, A. Presekala, E. Pramono, and R. F. Sari, “Measuring Quality of Services (QoS) of several forwarding strategies on Named Data Networking (NDN) using ndnSIM,” in *2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*. Phuket: IEEE, Nov. 2017, pp. 45–49.
- [273] D. Posch, B. Rainer, and H. Hellwagner, “Towards a context-aware forwarding plane in named data networking supporting QoS,” *Computer Communication Review*, vol. 47, no. 1, pp. 4–14, 2017.
- [274] W. Yang, J. Cao, and F. Wu, “Adaptive video streaming with scalable video coding using multipath QUIC,” in *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*. IEEE, 2021, pp. 1–7.
- [275] B. Banerjee, “A study of optimized caching and user mobility in wireless cache-enabled networks,” Master of Science in Communication, University of Alberta, 2018.
- [276] M. Alreshoodi and J. Woods, “Survey on QoE QoS correlation models for multimedia services,” *International Journal of Distributed and Parallel Systems (IJDPS)*, vol. 4, no. 3, pp. 53–72, 2013. [Online]. Available: <https://arxiv.org/ftp/arxiv/papers/1306/1306.0221.pdf>

- [277] S. Petrangeli, J. V. D. Hooft, T. Wauters, and F. D. Turck, "Quality of Experience-Centric management of adaptive video streaming services," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 2s, p. 31, 2018. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=3210485.3165266>
- [278] J. Auge, G. Carofiglio, G. Grassi, L. Muscariello, G. Pau, and X. Zeng, "MAP-Me: Managing anchor-less producer mobility in Content-Centric Networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 590–610, 2018. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8267132/https://hal.sorbonne-universite.fr/hal-01700786/document>
- [279] S. Dhara, A. Majidi, and S. Clarke, "Revving up VNDN: Efficient caching and forwarding by expanding content popularity perspective and mobility," *Computer Communications*, vol. 212, pp. 342–352, 2023.
- [280] Y. Zha, P. Cui, Y. Hu, L. Xue, J. Lan, and Y. Wang, "An NDN cache-optimization strategy based on dynamic popularity and replacement value," *Electronics (Switzerland)*, vol. 11, 10 2022.
- [281] D. Keum, J. Lim, and Y. B. Ko, "Trust based multipath qos routing protocol for mission-critical data transmission in tactical ad-hoc networks," *Sensors (Switzerland)*, vol. 20, pp. 1–15, 6 2020.
- [282] M. Ul Haque, K. Pawlikowski, A. Willig, and L. Bischofs, "Performance analysis of blind routing algorithms over content centric networking architecture," in *2012 International Conference on Computer and Communication Engineering, ICCCE 2012*, no. July 2012. — Kuala Lumpur: IEEE, Jun. 2012, pp. 922–927.
- [283] S. Awang Nor, R. Alubady, and W. Abduladeem Kamil, "Simulated performance of TCP, SCTP, DCCP and UDP protocols over 4G network," *Procedia computer science*, vol. 111, no. 2015, pp. 2–7, 2017. [Online]. Available: <http://dx.doi.org/10.1016/j.procs.2017.06.002>
- [284] N. N. Dao, A. T. Tran, N. H. Tu, T. T. Thanh, V. N. Q. Bao, and S. Cho, "A contemporary survey on live video streaming from a computation-driven perspective," *ACM Computing Surveys*, vol. 54, 11 2022.
- [285] B. Alahmri, S. Al-Ahmadi, and A. Belghith, "Efficient pooling and collaborative cache management for NDN/IoT networks," *IEEE access*, vol. 9, pp. 43 228–43 240, 2021.
- [286] M. Hassan and R. Jain, *High performance TCP / IP networking*. Upper Saddle River, New Jersey: Prentice Hall Upper Saddle River, NJ, 2003. [Online]. Available: [PrenticeHallUpperSaddleRiver,NJ](http://PrenticeHallUpperSaddleRiver,NJ)
- [287] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla, "Glomosim: A scalable network simulation environment," *Compare A Journal Of Comparative Education*, vol. 990027, no. 1999, p. 213, 1999.

- [288] Frederick J. Gravetter and Lori-Ann B. Forzano, *Research methods for the behavioral sciences*, sixth edition ed. 20 Channel Center Street, Boston, MA 02210, USA: Cengage Learning, 2018.
- [289] R. K. Jain, *Art of computer systems performance analysis: Techniques for experimental design measurements... simulation and modeling*. John Wiley, 2015. [Online]. Available: <https://www.wiley.com/en-us/The+Art+of+Computer+Systems+Performance+Analysis%3A+Techniques+for+Experimental+Design%2C+Measurement%2C+Simulation%2C+and+Modeling-p-9780471503361>
- [290] G. Zhang, B. Tang, P. Wang, Y. Wu, and X. Zhang, “Performance assessment of cache strategies in content centric network,” *Information Technology Journal*, vol. 12, no. 23, pp. 7083–7089, 2013.
- [291] G. Zhang, Y. Li, and T. Lin, “Caching in Information Centric Networking: A survey,” *Computer Networks*, vol. 57, no. 16, pp. 3128–3141, Nov. 2013.
- [292] R. Asokan and A. M. Natarajan, “An approach for reducing the end-to-end delay and increasing network lifetime in mobile Ad Hoc networks,” *Int J Inf Technol*, vol. 4, no. 2, pp. 121–127, 2008. [Online]. Available: <http://waset.org/publications/14010>
- [293] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, K. Claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh, “NSF Named Data Networking (NDN),” *NDN-0001, Palo Alto Research Center-PARC*, 2010. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.366.6736%7B%5C%7Drep=rep1%7B%5C%7Dtype=pdf>
- [294] F. Qaiser, K. Said Al Harthy, M. Hussain, J. Frnda, R. Amin, R. Gantassi, and M. D. Zakaria, “Classifications and Analysis of Caching Strategies in Information-Centric Networking for Modern Communication Systems,” *Engineering Reports*, vol. 7, no. 2, p. e70005, 2025.
- [295] S. Cao, Q. Zheng, Z. Zhan, Y. Yang, H. Lv, D. Zheng, and W. Zhang, “FlyCache: Recommendation-Driven Edge Caching Architecture for Full Life Cycle of Video Streaming,” *Digital Communications and Networks*, 2025.
- [296] M. K. Kumari, N. Tripathi, and P. Joshi, “ProxaDyn: A Proximity-aware Dynamic Caching Approach for Named Data Networks,” *IEEE Transactions on Network Science and Engineering*, 2025.
- [297] X. Zhang, “Cross-layer P2P traffic engineering in content-based networks,” Doctoral thesis, University of Surrey, 2015.
- [298] J. Lynch, “New study says by 2025, half of consumers under 32 won’t pay for cable-resistance is futile: The’cord-nevers’ cannot be stopped. Ed. by Adweek Inc,” p. 1, 2015.
- [299] Cisco, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update,” p. 14, 2017.

- [300] R. H. Weber, "Digital trade and E-commerce: challenges and opportunities of the Asia-Pacific regionalism," *Asian J. WTO and Int'l Health L aand Pol'y*, vol. 10, p. 321, 2015.
- [301] C. Li, W. Liu, L. Wang, M. Li, and K. Okamura, "Energy-efficient quality of service aware forwarding scheme for Content-Centric Networking," *Journal of Network and Computer Applications*, vol. 58, pp. 241–254, 2015.
- [302] Z. Li and G. Simon, "Time-shifted TV in content centric networks: The case for cooperative in-network caching," in *2011 IEEE International Conference on Communications (ICC)*. Kyoto: IEEE, Jun. 2011, pp. 1–6.
- [303] Y. Chen, E. Le Merrer, Z. Li, Y. Liu, and G. Simon, "OAZE: A network-friendly distributed zapping system for peer-to-peer IPTV," *Computer Networks*, vol. 56, no. 1, pp. 365–377, 2012.
- [304] Y. Li, H. Xie, Y. Wen, and Z. L. Zhang, "Coordinating in-network caching in content-centric networks: Model and analysis," in *Distributed Computing Systems (ICDCS) 2013 IEEE 33rd International Conference on*. Philadelphia, PA: IEEE, Jul. 2013, pp. 62–72.
- [305] Y. Chen, J. Leblet, and G. Simon, "On reducing the inter-AS traffic of box-powered CDN," in *2009 Proceedings of 18th International Conference on Computer Communications and Networks*. IEEE, 2009, pp. 1—6.
- [306] M. Rezazad and Y. C. Tay, "CCndnS: A strategy for spreading content and decoupling NDN caches," in *IFIP Networking Conference (IFIP Networking), 2015*. Toulouse: IEEE, May 2015, pp. 1–9.
- [307] R. S. Abbas, S. A. Nor, and A. Ahmad, "Coordination and Non-coordination Caching Placement Algorithms in Named Data Network for Video on Demand Services," *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 24, no. 2, pp. 14–21, Mar. 2022.
- [308] B. Berg, D. S. Berger, S. McAllister, I. Grosf, S. Gunasekar, J. Lu, M. Uhlar, J. Carrig, N. Beckmann, M. Harchol-Balter *et al.*, "The {CacheLib} caching engine: Design and experiences at scale," in *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, 2020, pp. 753–768.
- [309] S. He, W. Huang, J. Wang, J. Ren, Y. Huang, and Y. Zhang, "Cache-Enabled Coordinated Mobile Edge Network: Opportunities and Challenges," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 204–211, 2020.
- [310] Y. Yuan, X. Wang, and G. Bin, "Analysis of user behavior in a large-scale Internet video-on-demand(VoD) system," in *Proceedings of the 5th International Conference on Multimedia and Image Processing*. ACM, 2020, pp. 153–158.
- [311] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Modeling data transfer in content-centric networking," in *2011 23rd International Teletraffic Congress (ITC)*. San Francisco, CA, USA: IEEE, 2011, pp. 111–118. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/6038471>

- [312] M. Alipio, N. M. Tiglao, A. Grilo, F. Bokhari, U. Chaudhry, and S. Qureshi, "Cache-based transport protocols in wireless sensor networks: A survey and future directions," *Journal of Network and Computer Applications*, vol. 88, pp. 29–49, 2017.
- [313] I. Abdullahi, S. Hassan, and S. Arif, "Prospective use of bloom filter and muxing for Information Centric Network caching," *ARPJ Journal of Engineering and Applied Sciences*, vol. 10, no. 3, pp. 1169–1177, 2015.
- [314] J. Wang, H. Wu, and R. Wang, "A new reliability model in replication-based big data storage systems," *Journal of Parallel and Distributed Computing*, vol. 108, pp. 14–27, Oct. 2017.
- [315] Y. Thomas, G. Xylomenos, C. Tsilopoulos, and G. C. Polyzos, "Object-Oriented packet caching for ICN," in *Proceedings of the 2nd ACM Conference on Information-Centric Networking*. San Francisco, California: ACM, 2015, pp. 89–98.
- [316] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, jul 2014.
- [317] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox, "Information-Centric Networking: Seeing the forest for the trees," in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*. Cambridge: ACM, 2011, pp. 1–6.
- [318] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, "Understanding user behavior in large-scale video-on-demand systems," *Proceedings of the 2006 EuroSys Conference*, vol. 40, no. 4, pp. 333–344, 2007.
- [319] Z. Jia, P. Zhang, J. Huang, C. Lin, and J. C. S. Lui, "Modeling Hierarchical Caches in Content-Centric Networks," in *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, Nassau, Bahamas, Jul. 2013, pp. 1–7.
- [320] D. Huang, X. Zhang, W. Shi, M. Zheng, S. Jiang, and F. Qin, "LiU: Hiding Disk Access Latency for HPC Applications with a New SSD-Enabled Data Layout," in *2013 IEEE 21st International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems*. San Francisco, CA, USA: IEEE, Aug. 2013, pp. 111–120. [Online]. Available: <http://ieeexplore.ieee.org/document/6730754/>
- [321] M. Dehghan, W. Chu, P. Nain, D. Towsley, and Z. L. Zhang, "Sharing Cache Resources among Content Providers: A Utility-Based Approach," *IEEE/ACM Transactions on Networking*, vol. 27, no. 2, pp. 477–490, 2019.
- [322] A. Sharma, A. Venkataramani, and R. K. Sitaraman, "Distributing content simplifies ISP traffic engineering," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 1, pp. 229–242, 2013.

- [323] P. Fennell, “Extremes of XML,” *XML LONDON 2013*, pp. 80–86, Sep. 2013.
- [324] I. Sodagar, “The MPEG-DASH standard for multimedia streaming over the Internet,” *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, 2011.
- [325] Y. Liu, J. Geurts, J.-C. Point, S. Lederer, B. Rainer, C. Muller, C. Timmerer, and H. Hellwagner, “Dynamic Adaptive Streaming over CCN : A caching and overhead analysis,” in *Communications (ICC), 2013 IEEE International Conference on*. Budapest: IEEE, Jun. 2013, pp. 3629–3633.
- [326] R. Grandl, K. Su, and C. Westphal, “On the interaction of adaptive video streaming with Content-Centric Networking,” *arXiv preprint arXiv:1307.0794*, p. 8, 2013.
- [327] W. Li, S. M. A. Oteafy, and H. S. Hassanein, “Dynamic adaptive streaming over popularity-driven caching in Information-Centric Networks,” in *Communications (ICC), 2015 IEEE International Conference on*. London: IEEE, Jun. 2015, pp. 5747–5752.
- [328] A. Ramakrishnan, C. Westphal, and J. Saltarin, “Adaptive video streaming over CCN with network coding for seamless mobility,” in *2016 IEEE International Symposium on Multimedia (ISM)*. San Jose, CA: IEEE, Dec. 2016, pp. 238–242.
- [329] J. Samain, G. Carofiglio, M. Tortelli, and D. Rossi, “A simple yet effective network-assisted signal for enhanced DASH Quality of Experience,” in *Proceedings of the 28th ACM SIGMM Workshop on Network and Operating Systems Support for Digital Audio and Video*. Amsterdam: ACM, Jun. 2018, pp. 55–60.
- [330] D. Sisalem and A. Wolisz, “MLDA: a TCP-friendly congestion control framework for heterogeneous multicast environments,” in *2000 Eighth International Workshop on Quality of Service. IWQoS 2000 (Cat. No.00EX400)*. IEEE, 2001, pp. 65–74.
- [331] S. Puangpronpitag, “Design and performance evaluation of multicast congestion control for the Internet,” Doctoral Thesis, University of Leeds, Nov. 2003.
- [332] Z. Li, J. Li, Q. Wu, G. Tyson, and G. Xie, “A large-scale measurement and optimization of mobile live streaming services,” *IEEE Transactions on Mobile Computing*, vol. 22, no. 12, pp. 7294–7309, 2022.
- [333] O. Ayoub, F. Musumeci, M. Tornatore, and A. Pattavina, “Energy-efficient video-on-demand content caching and distribution in metro area networks,” *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 1, pp. 159–169, 2018.
- [334] J. Li, B. Liu, and H. Wu, “Energy-efficient in-network caching for content-centric networking,” *IEEE Communications Letters*, vol. 17, no. 4, pp. 797–800, 2013.

- [335] H. Sun, H. Zhang, W. Guan, X. Liu, H. Ma, and V. Leung, "Enhancing Low-Latency and High-Reliability Data Forwarding for In-Vehicle Time-Sensitive Networks," *IEEE Transactions on Vehicular Technology*, 2025.
- [336] S. M. A. Iqbal, Asaduzzaman, and M. M. Hoque, "A source-driven probabilistic forwarding and caching strategy in NDN and SDN-based NDN," *International Journal of Communication Systems*, vol. 35, no. 6, p. e5093, 2022.
- [337] Z. Zhang, "Performance Improvements of In-Network Caching in ICN-Based Networks," Ph.D. dissertation, Carleton University, 2019.
- [338] T.-T. Nguyen, L. Liebe, N.-Q. Tau, Y. Wu, J. Cheng, and D. Lee, "OCTOPINF: Workload-Aware Inference Serving for Edge Video Analytics," *arXiv preprint arXiv:2502.01277*, 2025.
- [339] T. Nunome and Y. Takada, "ICN/CCN Multi-View Video QoE with Cache Control Methods on a Tree Topology Network," *IEEE Consumer Electronics Magazine*, 2025.
- [340] R. Farahani, "Network-Assisted Delivery of Adaptive Video Streaming Services through CDN, SDN, and MEC," *arXiv preprint arXiv:2403.16951*, 2024.

