Efficient Caching with Network Partitioning in Content Centric Network

A thesis submitted to the

University of Petroleum and Energy Studies

For the Award of

Doctor of Philosophy

in

Computer Science and Engineering

By

Sumit Kumar

October 2021

Supervisor

Dr. Rajeev Tiwari

UPES

UNIVERSITY WITH A PURPOSE

SCHOOL OF COMPUTER SCIENCE

University of Petroleum and Energy Studies

Energy Acres, P.O. Bidholi via Prem Nagar,

Dehradun, 248007: Uttarakhand, India.

Efficient Caching with Network Partitioning in Content Centric Network

A thesis submitted to the
University of Petroleum and Energy Studies

For the Award of
Doctor of Philosophy
in
Computer Science and Engineering

By

Sumit Kumar
(SAP ID 500064227)

October 2021

Supervisor
Dr. Rajeev Tiwari
Sr. Associate Professor
School of Computer Science
University of Petroleum and Energy Studies

UPES
UNIVERSITY WITH A PURPOSE

SCHOOL OF COMPUTER SCIENCE
University of Petroleum and Energy Studies
Energy Acres, P.O. Bidholi via Prem Nagar,
Dehradun, 248007: Uttarakhand, India.

Declaration

I declare that the thesis entitled **"Efficient Caching with Network Partitioning in Content Centric Network"** has been prepared by me under the guidance of Dr. Rajeev Tiwari, Sr. Associate Professor of School of Computer Science, University of Petroleum & Energy Studies. No part of this thesis has formed the basis for the award of any degree or fellowship previously.

19/10/2021

Sumit Kumar
School of Computer Science
University of Petroleum & Energy Studies,
Bidholi via Prem Nagar, Dehradun, Uttarakhand, India
Date: October 19, 2021

Certificate

I certify that **Mr. Sumit Kumar** has prepared his thesis entitled **"Efficient Caching with Network Partitioning in Content Centric Network"**, for the award of PhD degree of the University of Petroleum & Energy Studies, under my guidance. He has carried out the work at the School of Computer Science, University of Petroleum & Energy Studies.

Rajeev Tiwari
19/10/2021

Supervisor

Dr. Rajeev Tiwari

Sr. Associate Professor

School of Computer Science

University of Petroleum & Energy Studies,

Bidholi via Prem Nagar, Dehradun, Uttarakhand, India

Date: October 19, 2021

ABSTRACT

Content-Centric Networking (CCN) become one of the prominent and emerging future Internet architecture. CCN provides efficient content distributions by retrieving the contents with their names. In CCN, the in-network content caching is an essential characteristic that largely influence the performance of the networks and Quality-of-Service (QoS) for requesters. The content caching capability of the network routers reduces the load from servers, content access delay and network traffic. Towards this, there is a need of efficient caching scheme to comprehensively utilize the available caching resources. In this thesis, two novel content caching schemes are proposed, named PDC (Popularity window and Distance-based Caching) and DPPCOP (Dynamic Partitioning and Popularity based Caching for Optimized Performance), for efficient utilization of the cache space. The PDC caching strategy takes autonomous caching decisions based on the content popularity and distance parameters. Using these heuristics, the scheme increases the caching probability for popular contents towards the edges of the network. Hence, the largely requested contents are accessed from the network with lesser delay and network traffic. However, a careful investigation of the PDC caching scheme later revealed that the scheme suffers from content redundancy issue in high content popularity distributions. Therefore, the proposed PDC scheme have further scope of improvement. For this, a novel DPPCOP caching scheme is proposed that dynamically partition the network and takes collaborative caching decisions without significant communication overhead. The DPPCOP scheme considers content popularity and distance-based parameters along with the partitioning information during content placements. For content evictions, the Least-Recently Used algorithm has been implemented in the PDC and DPPCOP strategies.

Extensive simulations are performed on the realistic network topology using ndnSIM simulation tool to examine the performance of the proposed caching solutions. The simulation results illustrate that the proposed PDC and DPPCOP caching schemes improves the network performance as compared to existing schemes based on the performance metrics such as cache hit ratio, average network hop count, delay and network traffic.

Keywords- Content-Centric Networking, Network partitioning, Content placement, Content popularity, Cache replacement, Dynamic popularity window, In-network caching, PDC, DPPCOP.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Dr. Rajeev Tiwari for the motivation and guidance during the Ph.D. and related research. It would never be forgotten. His immense knowledge and encouragement helped me in all the time during this journey and writing of this thesis.

Very special thanks to Dr. Sunil Rai, Vice Chancellor, Prof. D. K. Avasthi, Dean-R & D (UPES), Dr. Pankaj Kumar, Associate Dean- R & D (UPES) and Dr. Neelu Jyothi Ahuja, Cluster Head (Systemics) for their support and providing pleasant research environment in the University.

I am grateful for my mother, brother and lovely daughter whose never-ending care keep me motivated and confident. My success and accomplishment are due to their believe in me. Deepest thanks to Surbhi, my beloved wife. I am always thankful for her unconditional love throughout the journey of Ph.D.

Sumit Kumar

Table of Contents

Declaration **i**

Certificate **ii**

Abstract **iii**

Acknowledgements **v**

Table of Contents **vi**

List of Abbreviations **x**

List of Figures **xii**

List of Tables **xv**

List of Publications **xvi**

1 Introduction **1**

1.1 Overview of CCN-based architecture 4

1.2 Need for in-network caching in Content-Centric Networking . 9

1.2.1 Challenges for efficient utilization of in-network caching in CCN 10

1.3 Objective of research 13

1.4 Contributions of research 13

1.5 Organization of thesis 15

1.6 Summary 15

2 Literature Review **17**

2.1 Types of caching schemes 18

2.1.1 Probabilistic caching schemes 18

2.1.2 Graph-based caching schemes 22

2.1.3 Popularity-based caching schemes 25

2.1.4 Multi-attribute-based caching schemes 31

2.1.5 Collaboration and/or Partitioning-based content caching schemes 35

2.2 Gaps in Literature 41

2.3 Analysis 42

2.4 Summary 42

3 Proposed Content Popularity and Distance-based Caching Scheme (PDC) **44**

3.1 Problem Statement 44

3.2 System Model 46

3.2.1 Network Model 46

3.2.2 Dynamic model and assumptions 49

3.3 Design and operations of Dynamic Popularity Window . . . 51

3.4 Interest message processing 53

3.4.1 Content message processing 55

3.5 Cost analysis of the Dynamic Popularity Window in PDC strategy 57

3.6 Limitations of proposed PDC caching scheme 58

3.7 Summary 58

4 Dynamic Partitioning and Popularity based Caching for Optimized Performance (DPPCOP) **60**

4.1 Problem Statement 60

4.2 Proposed caching scheme: DPPCOP 61
4.3 Proposed Network Partitioning Scheme 62
4.4 An illustration of the proposed network partitioning scheme 65
4.5 Updated structure of Interest and Content messages 67
4.5.1 Modified structure of Interest message 67
4.5.2 Modified structure of Content message 68
4.5.3 Interest Message Handling Procedure 68
4.5.4 Content Message Handling Procedure 70
4.6 Simulation parameters and setup 72
4.6.1 Simulation parameters for PDC caching scheme . . . 73
4.6.2 Simulation parameters for DPPCOP caching scheme 74
4.7 Evaluation parameters & Criteria 75
4.7.1 Cache hit ratio 76
4.7.2 Average network hop count 77
4.7.3 Average network delay 77
4.7.4 Average network traffic 77
4.8 Summary 78

5 Result Discussions and Analysis 79

5.1 Result Discussions and Analysis of the Proposed PDC Caching Scheme 79
5.1.1 Effect on cache hit ratio (PDC caching scheme) . . . 80
5.1.2 Effect on average network hop count (PDC caching scheme) 82
5.1.3 Effect on average network delay (PDC caching scheme) 84
5.1.4 Effect on average network traffic (PDC caching scheme) 84
5.2 Result Discussions and Analysis of the Proposed DPPCOP Caching Scheme 88
5.2.1 Effect on cache hit ratio (DPPCOP caching scheme) 88

5.2.2 Effect on average network hop count (DPPCOP caching scheme) 89
5.2.3 Effect on average network delay (DPPCOP caching scheme) 92
5.2.4 Effect on average network traffic (DPPCOP caching scheme) 93
5.2.5 Discussion 94
5.3 Summary 96

6 Statistical validations 97

6.1 Two-tailed T-Test with unequal variance 97
6.2 Null Hypothesis and alternative hypothesis 98
6.3 Statistical validation of the performance of PDC caching scheme 99
6.4 Statistical validation of the performance of DPPCOP caching scheme 100
6.5 Summary 100

7 Conclusion and future research 102

7.1 Contributions 102
7.2 Future Scope 104

Bibliography 106

List of Abbreviations

CCN-Content-Centric Networking.

PDC- Popularity window and Distance-based Caching.

DPPCOP- Dynamic Partitioning and Popularity based Caching for Optimized Performance.

ARPANET- Advanced Research Projects Agency Network.

WAN- Wide Area Network.

IP- Internet Protocol).

URL- Uniform Resource Locator.

DNS- Domain Name Server.

DHCP- Dynamic Host Configuration Protocol.

TCP- Transmission Control Protocol.

ICN- Information-Centric Networking.

PARC- Palo Alto Research Center.

CS- Content Store.

FIB- Forwarding Information Base.

PIT- Pending Interest Table.

LCE- Leave-Copy Everywhere.

LRU- Least Recently Used.

FIFO- First In First Out.

LFU- Least Frequently Used.

QoS- Quality-of-Service.

LCD- Leave Copy Down.

MCD- Move Copy Down.

DC- Degree Centrality.

SC- Stress Centrality.

BC- Betweeness Centrality.

CC- Closeness Centrality.

GC- Graph Centrality.

CMBA- Centrality-measure based algorithm.

MPC- Most-Popular Content Caching.

FGPC- Fine-Grained Popularity-based Caching.

MAGIC- MAx-Gain In-network Caching.

DAG- Directed Acyclic Graph.

CPCCS- Compound Popular Content Caching Strategy.

MAC- Multi-Attribute Content Caching.

CSCPUL- Cache Scheme based on Content Popularity and User Location.

CPUL- Content Popularity and User Location.

NCPP- Node-Content Pass Probability.

CPNDD- Content Placement based on Normalized Node Degree and Distance.

VDHT- Virtual Distributed Hash Table.

PUC- Packet Update Caching.

HCC- Hierarchical Clustering-based Caching.

MANET- Mobile Ad-hoc Networks.

WSN- Wireless Sensor Networks.

VANET- Vehicular Ad Hoc Networks.

CBCMF- Cluster Based Content Management Framework.

CSDD- Collaborative Caching Strategy Distance and Node Degree.

CSR- Cluster-based Scalable Router.

WSS- Within-cluster Sum of Squares.

US- United States.

STU- Simulation Time Unit.

IoT- Internet-of-Things.

RSU- Road-Side Units.

List of Figures

1.1 An illustration of components in CCN with requesters, caching routers and content server with states of PIT, FIB and CS . . . 5

1.2 Conventional Content message processing . . . 8

2.1 Taxonomy of content caching schemes in Content-Centric networking *(Part-1/2)* . . . 19

2.2 Taxonomy of content caching schemes in Content-Centric networking *(Part-2/2)* . . . 20

3.1 An illustration of network topology in CCN . . . 48

3.2 Rank of 80-th percentile popular content for different catalog sizes according to Zipf distribution vs value of Zipf exponent parameter . . . 50

3.3 Functionality of Popularity Window in each in-network router 52

4.1 Partial illustration of Abilene network topology . . . 65

4.2 Illustration of partitioning in the partial Abilene network topology . . . 66

4.3 Partitions for the network mentioned in Fig. 4.1 . . . 67

4.4 Computation of (T_R) in the DPPCOP caching scheme . . . 76

5.1 In-network cache hit-ratio of caching schemes with C_{size}=50, $|D| = 5000$, $\alpha = 0.7$ and $\lambda = 50/second$ . . . 80

5.2 In-network cache hit-ratio of caching schemes with C_{size}=100, $|D| = 5000$, $\alpha = 0.7$ and $\lambda = 50/second$ 81
5.3 Average network hop-count of caching schemes with C_{size}=50, $|D| = 5000$, $\alpha = 0.7$ and $\lambda = 50/second$ 82
5.4 Average network hop-count of caching schemes with C_{size}=100, $|D| = 5000$, $\alpha = 0.7$ and $\lambda = 50/second$ 83
5.5 Average network delay (in micro-seconds) with C_{size}=50, $|D| = 5000$, $\alpha = 0.7$ and $\lambda = 50/second$ 85
5.6 Average network delay (in micro-seconds) with C_{size}=100, $|D| = 5000$, $\alpha = 0.7$ and $\lambda = 50/second$ 85
5.7 Average network traffic (in KB/second) with C_{size}=50, $|D| = 5000$, $\alpha = 0.7$ and $\lambda = 50/second$ 86
5.8 Average network traffic (in KB/second) with C_{size}=100, $|D| = 5000$, $\alpha = 0.7$ and $\lambda = 50/second$ 87
5.9 Average network hit ratio with C_{size}=50, $|D| = 5000$, $\alpha = 0.8$ and $\lambda = 50/second$ 89
5.10 Average network hit ratio with C_{size}=50, $|D| = 5000$, $\alpha = 1.1$ and $\lambda = 50/second$ 90
5.11 Average network hop-count with C_{size}=50, $|D| = 5000$, $\alpha = 0.8$ and $\lambda = 50/second$ 91
5.12 Average network hop-count with C_{size}=50, $|D| = 5000$, $\alpha = 1.1$ and $\lambda = 50/second$ 91
5.13 Average network delay (in microsecond) with C_{size}=50, $|D| = 5000$, $\alpha = 0.8$ and $\lambda = 50/second$ 92
5.14 Average network delay (in microsecond) with C_{size}=50, $|D| = 5000$, $\alpha = 1.1$ and $\lambda = 50/second$ 93
5.15 Average network traffic (in KB/second) with C_{size}=50, $|D| = 5000$, $\alpha = 0.8$ and $\lambda = 50/second$ 94

5.16 Average network traffic (in KB/second) with C_{size}=50, $|D| = 5000$, $\alpha = 1.1$ and $\lambda = 50/second$ 95

List of Tables

3.1 Variables definitions 47

4.1 Simulation parameters values 74

4.2 Simulation parameters values 75

5.1 Gain of proposed caching scheme on existing schemes based on cache hit ratio 81

5.2 Average hop count reduction in the proposed PDC caching scheme from existing schemes 84

5.3 Average network delay reduction in the proposed PDC caching scheme from existing schemes 86

5.4 Average network traffic reduction in the proposed PDC caching scheme from existing schemes 87

6.1 Statistical validations of the performance of PDC over existing peer scheme: Hit ratio and average network hop count 99

6.2 Statistical validations of the performance of DPPCOP over existing peer scheme: Average network delay and network traffic 100

6.3 Statistical validations of the performance of DPPCOP over existing peer scheme: Hit ratio and average network hop count 101

6.4 Statistical validations of the performance of DPPCOP over existing peer scheme: Average network delay and network traffic 101

List of Publications

Journals

[1]. Sumit Kumar and Rajeev Tiwari. "Optimized content centric networking for future Internet: dynamic popularity window based caching scheme." *Computer Networks* 179 (2020): 107434. (SCIE, Impact Factor-4.474), (Published). https://doi.org/10.1016/j.comnet.2020.107434

[2]. Sumit Kumar and Rajeev Tiwari. "An efficient content placement scheme based on normalized node degree in content centric networking." Cluster Computing 24 (2) (2021): 1277-1291. (SCIE, Impact Factor-1.809), (Published). https://doi.org/10.1007/s10586-020-03185-0

[3]. Sumit Kumar and Rajeev Tiwari. "Dynamic popularity window and distance-based efficient caching for fast content delivery applications in CCN." Engineering Science and Technology, an International Journal 24 (3) (2021): 829-837. (SCIE, Impact Factor-4.36), (Published). https://doi.org/10.1016/j.jestch.2020.12.018

[4]. Sumit Kumar, Rajeev Tiwari, Sergei Kozlov, and Joel JPC Rodrigues. "Minimizing delay in content-centric networks using heuristics-based in-network caching." Cluster Computing (2021): 1-15. (SCIE, Impact Factor-1.809), (Published). https://doi.org/10.1007/s10586-021-03405-1

[5]. Sumit Kumar and Rajeev Tiwari. "Optimization of Content-Centric Networking based IoT systems using partitioning-heuristics in-network caching." Journal of Ambient Intelligence and Humanized Computing (2021). (SCIE, Impact Factor-7.104), (Accepted).

Conferences

[1]. Sumit Kumar, Rajeev Tiwari, Mohammad S. Obaidat, Neeraj Kumar, and Kuei-Fang Hsiao. "CPNDD: Content placement approach in content centric networking." In ICC 2020-2020 IEEE International Conference on Communications (ICC), pp. 1-6. IEEE, 2020. (SCOPUS), (Published). https://doi.org/10.1109/ICC40277.2020.9149025

Chapter 1

Introduction

A computer network is a collection of interconnected autonomous computers which is used to share resources and information. The computers and the devices are termed as interconnected if they can exchange information with each other [1, 2]. The evolution of computer networks started in the late 1960s with ARPANET (Advanced Research Projects Agency Network) by U. S. Department of Defense. It serves as the starting point of the most widely used WAN (Wide Area Network), presently known as Internet [3, 4]. The purpose of the Internet is to provide communication support, data sharing, resource sharing, data security, and administration, etc.

The current IP (Internet Protocol)-oriented Internet architecture is based on host-to-host communication [5]. When a requester requires a content in the IP-based system, then it needs to send a request message to a server that stores the copy of the contents. To send the request message, the user first requires the IP address or URL (Uniform Resource Locator) of the server. A DNS (Domain Name Server) maps URL to IP address [6]. Therefore, whenever the user requires any content, they first need to search for the IP address of the server which holds the content. Then, the server transmits the content back to the requester using the destination IP address. This IP-based content retrieval architecture suffers from the following issues:

1. Host-based protocol and inefficient mobility support: To provide communication between users for content delivery, each user has a unique IP address, which is managed by the implementation mechanisms. This IP-handling mechanism becomes worse in mobile networks [7]. Although TCP provides mobile IP support [8] for these networks but the requirement of mobile IP is not clear, as most of the hosts are either mobile phones or laptops. In mobile phone communication, the mobility issue has been solved at the data link layer and mobile IP does not help in this environment. For laptops, mobility issues are largely handled using DHCP (Dynamic Host Configuration Protocol) with applications such as email and instant messaging by simply reconnecting to the server each time the IP address changes [9]. The DHCP protocol is managed either by a centralized server or by distributed architecture. However, both protocols suffer from frequent session management and route recalculations [10].

2. Inefficient congestion control: TCP's (Transmission Control Protocol) congestion control mechanism checks the network repeatedly to see the present amount of traffic and determine overload in the network after packet loss. Generally, the speed of content transmission in IP networks increases at a slow rate until a packet loss occurs. After a packet loss, the transmission rate decreases by half and again increases slowly till next occurrence of packet loss. This mechanism works well in the intial days of Internet but presently the requirements are changed [7]. The present Internet is more heterogeneous where TCP's current congestion protocols became inefficient. However, many researchers work on this problem, and various congestion control mechanisms are proposed [11,12] but these schemes only involve the end-user systems and not the intermediate routers. Therefore, the congestion cannot be controlled using the network routers.

3. Server load balancing issue: If a user requires content from the server, then it first needs to search the IP address of that server. Subsequently, the user creates a request and forwards it to the server via intermediate routers. At each intermediate router, the IP routing table [13] is consulted to find the next suitable hop. When the server receives the request, it prepares a reply message and transmits it towards the IP address of the user through intermediate routers. After the content retrieval, if any other user also needs the same content then the request again follows the same mechanism as discussed for the previous request [14]. Therefore, every time only the server can respond to the request. If the content is very popular and requested frequently by multiple users in the network, then the server becomes overloaded and network resources are not fully utilized due to high network redundancy.

The enormous growth of the Internet and its applications have generated vast new requirements from the IP architecture such as big data analysis, smart cities, smart transportation systems, automated toll tax systems, etc [15–17]. The initial Internet architecture was not designed to address these needs and provided patch-based solutions for these kind applications that increased the complexity of the system. In addition, these patches also proved to be only temporary solutions for the issues and various rising requirements cannot be handled by the present Internet [18].

In recent years, the network designing community has proposed several architectures to alleviate the above-mentioned limitations from the modern IP-based network environment and for the future of the Internet [19–22]. Presently, to a greater extent, the users are interested in retrieving the required content (data) with lesser latency rather than finding the specific host in the network. Therefore, the paradigm is shifted from the current host-centric architecture to the content-centric network design. Towards this, the Information-Centric Networking (ICN) has become a widely recognized and promising architecture to meet the requirements of future Internet [20, 21]. One of the promising architectures under the umbrella of ICN project is the Content-Centric Networking (CCN) which is developed by Van Jacobson and begin as a project under Palo Alto Research Center (PARC) [23]. CCN shifts Internet usage from the host-to-host-based communication paradigm to the receiver-driven content access mechanism. Hence, the content can not be received until it is requested from the receiver [24]. The CCN decouples the content from the hosts and each content has a unique name in the network [25]. During content access, the name of stored contents is matched with the requested content without focusing on the location. With this decoupling, the content can be accessed from anywhere in the network that provides timely retrieval of the content by the requesters [26].

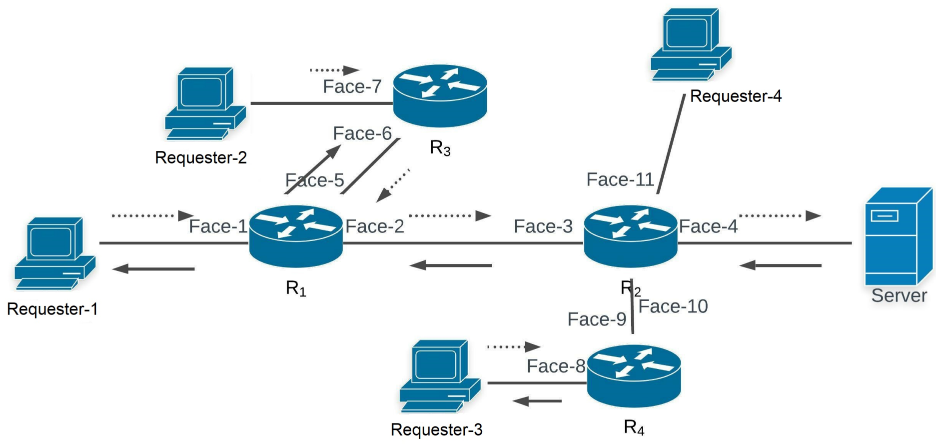
1.1 Overview of CCN-based architecture

As shown in Fig. 1.1, the CCN have the following basic components in the network [27]:

Content servers: The content servers are the nodes that publish named contents in the network. These nodes work as a sink for Interest messages and have a copy of all contents that can be required by the requester(s). In other words, the requested content is always found on the server.

Network routers: The network routers are the nodes that are capable of forwarding the incoming Content message towards the requester(s) in multi-hop communication. These nodes can also cache the incoming content. The requested content is retrieved from the network router if it has a copy of matching content.

Content requesters/consumers: These nodes generate the Interest messages to access contents in the network. The content requesters are the end-recipient of the Content message and they can be smartphones, laptops, desktops, sensors, etc.



Steps (1): Router R_1 tables after receiving INTEREST message from Requester

FIB	
Content	Out-Face
D_1	Face-2
PIT	
Content	Out-Face
CS	
Content	Payload

(2): Router R_1 tables after forwarding INTEREST message to Router R_2

FIB	
Content	Out-Face
D_1	Face-2
PIT	
Content	Out-Face
D_1	Face-1
CS	
Content	Payload

(6): Router R_1 tables after forwarding DATA message to requeter

FIB	
Content	Out-Face
D_1	Face-2
PIT	
Content	Out-Face
CS	
Content	Payload
D_1	---

(3): Router R_2 tables after receiving INTEREST message from Router R_1

FIB	
Content	Out-Face
D_1	Face-4
PIT	
Content	Out-Face
CS	
Content	Payload

(4): Router R_2 tables after forwarding INTEREST message to server

FIB	
Content	Out-Face
D_1	Face-4
PIT	
Content	Out-Face
D_1	Face-3
CS	
Content	Payload

(5): Router R_2 tables after forwarding DATA message to Router R_1

FIB	
Content	Out-Face
D_1	Face-4
PIT	
Content	Out-Face
CS	
Content	Payload
D_1	---

Figure 1.1: An illustration of components in CCN with requesters, caching routers and content server with states of PIT, FIB and CS

The communication in CCN is driven by the requester of content. The request is transmitted in the network using the Interest message and the corresponding data is retrieved in the form of Content message. The requester broadcasts the Interest message to its adjacent routers. Any intermediate router/server which is receiving the Interest message and having the requested data can respond with the matching Content message. In CCN, a Content message is transmitted only in reply to the Interest message. Since, Interest and Content messages carry a unique content name, the intermediate routers that are interested in the content, can share the transmission path using standard multicast suppression mechanisms [28].

To support content name-based communication in the CCN, the following data structures are implemented on the network nodes: Content Store (CS), Forwarding Information Base (FIB), and Pending Interest Table (PIT) [29]. The CS is the memory buffer of the network routers. Since the Content messages have unique identification and self-authentication in CCN, these messages can be useful for many requesters that are requesting the same content. To maximize the content sharing and the minimization of upstream bandwidth requirements, the Content messages are cached in the in-network routers. Therefore, instead of satisfying each Interest message at the server, several requests can also be handled at the in-network routers that have cached the matching content. It also reduces the content retrieval delay for the requesters as the requester gets the content from the closest available router that holds the requested content. The FIB contains the information of upstream faces to forward the Interest messages. It stores the information of potential sources for the requested content and allows the Interest message to be forwarded to one or more upstream faces simultaneously and searching of the content in parallel.

The PIT keeps the records of those Interest messages which are transmitted to the upstream faces towards the content source and the matching Content message is not yet received. Using PIT, the incoming Content message is forwarded downstream towards the requester(s). In CCN, the Interest messages are routed exclusively in the network and the Interest message also provides a trail for the matching Content message to follow the reverse path using PIT entries. Hence, the Content message is always routed through the “bread-crumbs” of the Interest message. If the requested content is not found within a defined duration in the network, then the Interest message entries of PIT are eventually timed out. Subsequently, the requester needs to re-express the request for the content, if it still requires that content.

When an Interest message reaches to the network router, the router begins searching for the matching content name in the cache (CS). If the content is matched (cache hit), then router (*content provider*) prepares a Content message and sends it to the requester using the "bread-crumbs" of the Interest message. If the content does not exist in the CS, then PIT is preferred for searching. If there is an exact match found in the PIT for the requested content then the information of its reaching face is appended with the stored requesting faces in the PIT. It indicates that the Interest message is already forwarded to the upstream routers and the newly received Interest message is discarded. The PIT entry ensures that when the matching Content message arrives, its copies would be sent to all the faces that are mentioned in the PIT records. Alternatively, if no match is found for the Interest message in the PIT, then FIB is searched to forward the Interest message towards all those upstream routers that may have the required content. After Interest message forwarding, its entry is created in the PIT along with its arrival face. If matching information is not there in FIB for message forwarding, then the Interest message is disposed from the network because the router is not having matching content and also does not know where to find it.

The $step-1$ in Fig. 1.1 illustrated that the $Requester-1$ prepares an Interest message for the content D_1 and forward it to its adjacent router R_1. When this message reaches to the router R_1, R_1 explores the cache for the matching content. As shown in the Figure, when content is not found, R_1 searches its FIB to transmit Interest message towards the upstream router using $(Face - 2)$ as shown in $step - 1$ in the Fig. 1.1. During forwarding of the request, R_1 maintains an entry for the forwarded Interest message in its PIT (shown in $step - 2$). A similar forwarding operation is performed by router R_2 on CS miss and it also creates a record in its PIT for the forwarded Interest message (shown in $step - 3$ and $step - 4$).

The processing of the incoming Content message is comparatively easy from the Interest message as the Content message travel the string of PIT entries for reaching to its requester(s). During Content message arrival, the router matches the name of received content in its CS. If an entry matches, it means that content is duplicate and hence, it is discarded. The processing of the Content message is shown in Fig. 1.2. As demonstrated, if no matching entry is found in the PIT, then it means that the incoming content is not requested and is discarded. Otherwise, the information of its incoming face is stored in the FIB for Interest message forwarding in the future.

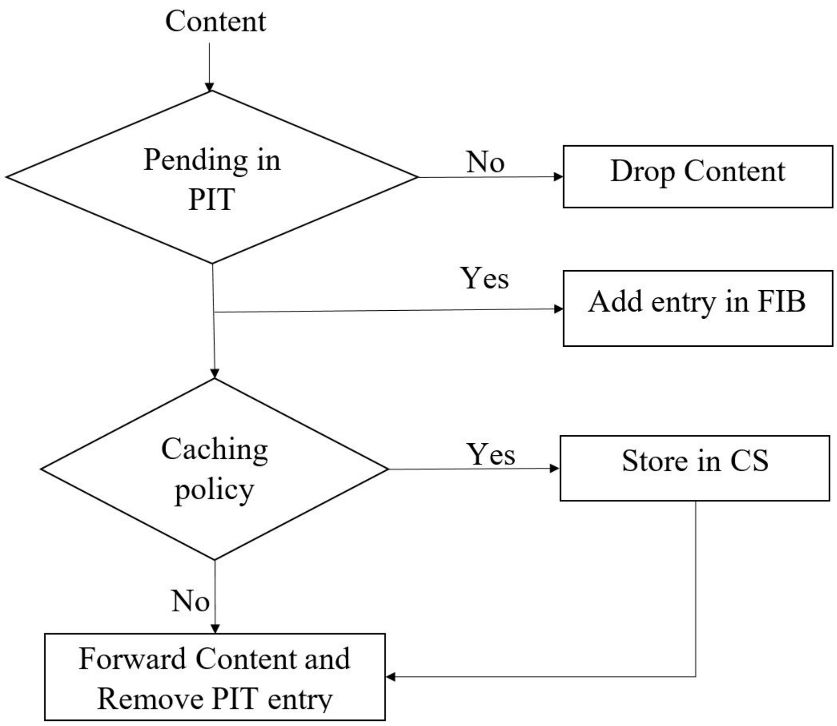


Figure 1.2: Conventional Content message processing

The CCN model allows the network routers to cache the incoming contents in their CS. All the routers can place the content in their CS based on the resource availability and the content cache mechanism. Independent from the caching decisions, the Content message is then forwarded on all the faces that match with its name in the PIT. Subsequently, the corresponding PIT entries are removed after Content forwarding to avoid repetition of Content message forwarding in the network.

The Content message forwarding and caching mechanism is illustrated in Fig. 1.1. When *server* forward the requested content D_1 to router R_2 in the reverse path, R_2 checks its PIT for the pending Interest message for D_1. On finding a request for D_1, R_2 forwards the content D_1 on $face-3$ towards router R_1 and cache the content copy in its CS as shown in $step-5$ of Fig. 1.1. Similar content caching and forwarding operations are performed in router R_1, which places a copy of D_1 in its CS as shown in $step-6$ in Fig. 1.1.

1.2 Need for in-network caching in Content-Centric Networking

In-network content caching is a fundamental building component in CCN. Caching becomes a modern research area because it is integrated as a key architectural element of CCN as compared to the overlay approach in the current IP-based Internet [30,31]. Every network router has content caching capability in the CCN and they can cache the contents which are transmitted from the server. The in-network content caching has many advantages for CCN as discussed below:

1) Improved QoS for requesters/consumers: The aim of in-network caching is to increase network performance and QoS for requesters by efficiently disseminating the content in the network. With the content caching capability of the network routers, the requester can access the required data from nearby routers instead of retrieving every content from the server. Therefore, caching feature minimizes content retrieval delay.

2) Reduction in network traffic and congestion: Since, the network routers are capable of caching the contents and serve the contents to requesters, the Interest messages need not be necessarily routed to the content server. Therefore, accessing the copy of the content from network routers reduces the traffic and the congestion in the network.

3)Server load reduction: With in-network caching, if a matching content exists in the CS of a router, then the routers can directly respond to the corresponding Interest messages. It means that only those Interest messages are served by the content server for which the content is not found in the cache of intermediate routers. Hence, content caching reduces the load from the server.

1.2.1 Challenges for efficient utilization of in-network caching in CCN

As discussed in the previous section, in-network content caching improves the network performance in CCN. Therefore, it is crucial to effectively utilize the available in-network caching capacity to increase QoS for the requesters (users). In CCN, a content caching scheme needs to address the following challenges during caching decisions: content placement strategy and content replacement strategy [32, 33].

The content placement scheme takes decisions regarding two challenges: selection of appropriate content for caching in CCN routers and selection of suitable router in the network for content storage [34]. Hence, the content placement scheme takes decisions regarding the selection of the content and location for caching in the network. The most cited and traditional content placement strategy of CCN is Leave-Copy Everywhere (LCE). In LCE, each content is placed in the cache of all intermediate routers along the delivery path without any discrimination.

The content replacement scheme is used when the cache reaches its maximum capacity and the new content needs to be cached. The content replacement strategy evicts the older content from the CS of the router to cache the incoming content. Least Recently Used (LRU) is one of the commonly implemented replacement schemes where the least recently queried content is evicted from the CS to cache new arriving content [21, 35–37]. The other widely used cache replacement schemes are First In First Out (FIFO) and Least Frequently Used (LFU) policies [38]. During FIFO replacement operations, the routers replaces the contents in the order of their arrival. In the LFU replacement strategy, the content that has been accessed least frequently is discarded to cache incoming content [39].

In realistic networks, the caching capacities of in-network routers are very small to cache all the distinguishable contents available in the network [40]. The traditional LCE caching scheme [23] performs unrestrained caching operations as each content is placed in all the intermediate network routers that exist in the delivery route. This excessive content caching increases the content redundancy and caching operations which leads to performance degradation in the network [41]. Hence, the LCE caching scheme does not utilize the available in-network caching capacities efficiently and also causes a high content replacement rate. In this direction, various existing research works have shown that caching only a subset of contents on selected routers can significantly increase the network performance in contrast to the traditional caching scheme [42–45].

In CCN, the caching mechanisms are widely classified as on-path and off-path caching schemes [46, 47]. The on-path caching schemes place the replica of content in the routers which exist in the content retrieval route and consider recommended content placement strategy for caching decisions [48]. Contrarily, the off-path caching mechanisms can cache the requested content in any network router which may or may not exists in the content delivery route. Due to this, the off-path caching schemes suffer from high computational complexity and increased network traffic/congestion in contrast to the on-path schemes. Therefore, the on-path caching strategies become more suitable for content placements and are vastly deployed in the Content-Centric Networking [29].

When the network routers take autonomous caching decisions, the communication overhead reduces significantly. However, in autonomous caching schemes, the same content might be excessively placed in many intermediate on-path routers [49, 50] due to no-coordination among the network nodes. Due to this, autonomous caching decisions increase content duplication in the network and the requests for other contents need to be served from the server. It limits the network performance substantially as the cache hit rate decreases and consequently, the content retrieval delay increases. Therefore, there is a need for collaboration among the network routers to minimize unrestrained content placement/replacement operations and increasing the cache hit rate in the network. Moreover, the investigation of caching strategies is also required to develop a CCN-based environment that is scalable to the increasing needs of its applications. Specifically, a mechanism is required that resolve the content placement issues by jointly considering the network and content parameters. These parameters can be node degree, content popularity, cache capacity, number of cache replacements, the overhead to fetch the identical content from the server, etc. The proposed research works also need to formulate an efficient caching scheme for CCN with a network partitioning scheme for collaboration among routers. Additionally, the proposed caching solution should increase cache hit rate, reduce network traffic, hop-count, and content retrieval delay in the network.

1.3 Objective of research

After analyzing the research gaps in the existing works related to in-network caching in CCN, the following objectives have been framed. The objective of the research is to design and implement an efficient caching scheme with network partitioning for Content Centric network.

Sub-objectives of research:

1. Design and implement a novel network-partitioning algorithm in CCN.
2. Design and implement an efficient content placement algorithm for CCN using simulator.
3. Validation of proposed technique with existing standard peer techniques from literature on output parameters like hit ratio, latency, network traffic, etc.

1.4 Contributions of research

The focus of the proposed research work is towards the efficient utilization of the available in-network caching capacity to improve QoS for the requesters and achieving improved network performance. In this direction, the contribution of the present research work is two-folded; dynamic partitioning of the network and a novel caching scheme for effective content placement and replacement decisions. In the initial phase, the joint effect of various parameters have been examined on the caching performance such as node degree centrality, content popularity and hop count attributes. Then, the proposed caching scheme dynamically partition the network into the optimal number of partitions based on the bandwidth and hop count parameters. The network partitioning is performed to reduce the content placement/replacement operations and content redundancy. In the proposed caching scheme, each partition controls its caching operations collaboratively. When the content provider and the requester exist in the same partition, then the content is not cached in the intermediate routers to minimize computational delay and caching operations. When the content provider and its requester belong to different partitions then at most one copy of the content is placed in each intermediate partition (on-path routers collaborate with each other) according to the proposed caching scheme. Using these heuristics, the content redundancy is reduced within partitions and subsequently, the caching performance improves in the whole network.

For efficient caching decisions, the proposed work jointly considers the content popularity and the distance-based parameters along with network partitioning information. In the proposed caching strategy, the content placement probability increases for the popular content as it traverses from the provider to the requester. Therefore, the proposed scheme places the popular contents near the requesters with a higher probability. It has been argued that it would improve the QoS for the requesters/consumers and the network performance as the cache hit ratio would increase that lead to lesser network traffic and content retrieval delay.

1.5 Organization of thesis

The proposed research work is discussed in six chapters including the introduction chapter.

Chapter 2 elaborates the existing content caching schemes for CCN. The chapter discusses the gaps in the selection of suitable contents and routers during content placement/replacement decisions. The identified gaps motivated for the proposed research work.

The system model and two novel content caching schemes are presented in Chapter 3 and Chapter 4. The chapter includes the algorithm for dynamic network partitioning and the $Interest/Content$ message processing for efficient utilization of the available caching resources.

Chapter 5 presents the performance evaluation of the proposed caching scheme using extensive simulations on realistic network topology. This chapter examines the performance of the proposed caching solutions along with the existing caching strategies on metrics such as network hit ratio, hop count, average network delay, and network traffic metrics.

The performance gain achieved by the proposed caching scheme is statistically validated in Chapter 6.

Finally, Chapter 7 summarizes the contributions of the proposed research work. The chapter also discusses the future scope of proposed work in CCN.

1.6 Summary

This chapter discusses the significance of in-network content caching in the CCN and addresses the major concerns in the efficient utilization of the caching resources. The section analyzes the need for an efficient caching scheme to improve network performance and QoS for the requesters. The next section elaborates the existing caching solutions given by various researchers for Content-Centric Networking.

Chapter 2

Literature Review

In CCN, when a content is requested and no on-path router has a matching cache entry for the requested content then it is retrieved from the content server [51]. During content message forwarding, the content is placed in the intermediate routers and later, the neighboring routers/requesters can access this content at a high speed with lesser delay and improved efficiency. This approach of content retrieval is known as a pull-based access mechanism in which the content is transmitted only when it is requested by the consumer [52].

Since on-path caching schemes place the content in those routers that exist in the delivery route, these schemes ended with the caching decisions in a very limited time without additional communication overhead. This improves the propagation delay for the Content messages. Moreover, these schemes do not introduce more network traffic and computational complexity as compared to off-path caching schemes [53]. Generally, these in-network caching decisions are valid for short time durations and increases the QoS for requesters [54]. This section discusses existing on-path caching schemes that suggest different heuristics for the content placement/replacement operations and the selection of network routers during caching decisions for improved network performance.

2.1 Types of caching schemes

The on-path content caching strategies are broadly classified into five categories: Probabilistic caching [23, 55, 56], Graph-based [57–61], Popularity-based [35, 62, 63], Multi-attribute-based [64–68] and Collaboration/Partitioning-based caching policies [37, 69–73]. This section discuss the noteworthy caching strategies that are falling under these categories.

2.1.1 Probabilistic caching schemes

The probabilistic caching solutions consider the probability metric for placing the contents in on-path network routers. This probability is determined based on the underlying mechanism in the caching strategy. Generally, these schemes take autonomous caching decisions without any cooperation among routers. Some of the probability-based caching schemes are described as follows.

The traditional CCN caching scheme i.e. LCE [23] places the content on each intermediate router in the delivery path with 100% probability. The placement of content copy on every intermediate router improves the QoS for the requesters and reduces the server load as the content is accessed from the nearby routers. However, excessive caching operations result in very high content redundancy and resource consumptions. Additionally, the routers which are close to the server are more burdened than those routers that are near to the requesters during caching operations based on the LCE strategy. Therefore, LCE scheme is generally used as the traditional benchmark strategy to analyze the performance gain of newer caching solutions.

A random fixed-probability caching algorithm is proposed in [55]. The design of the algorithm is targeted on a pull-based application environment

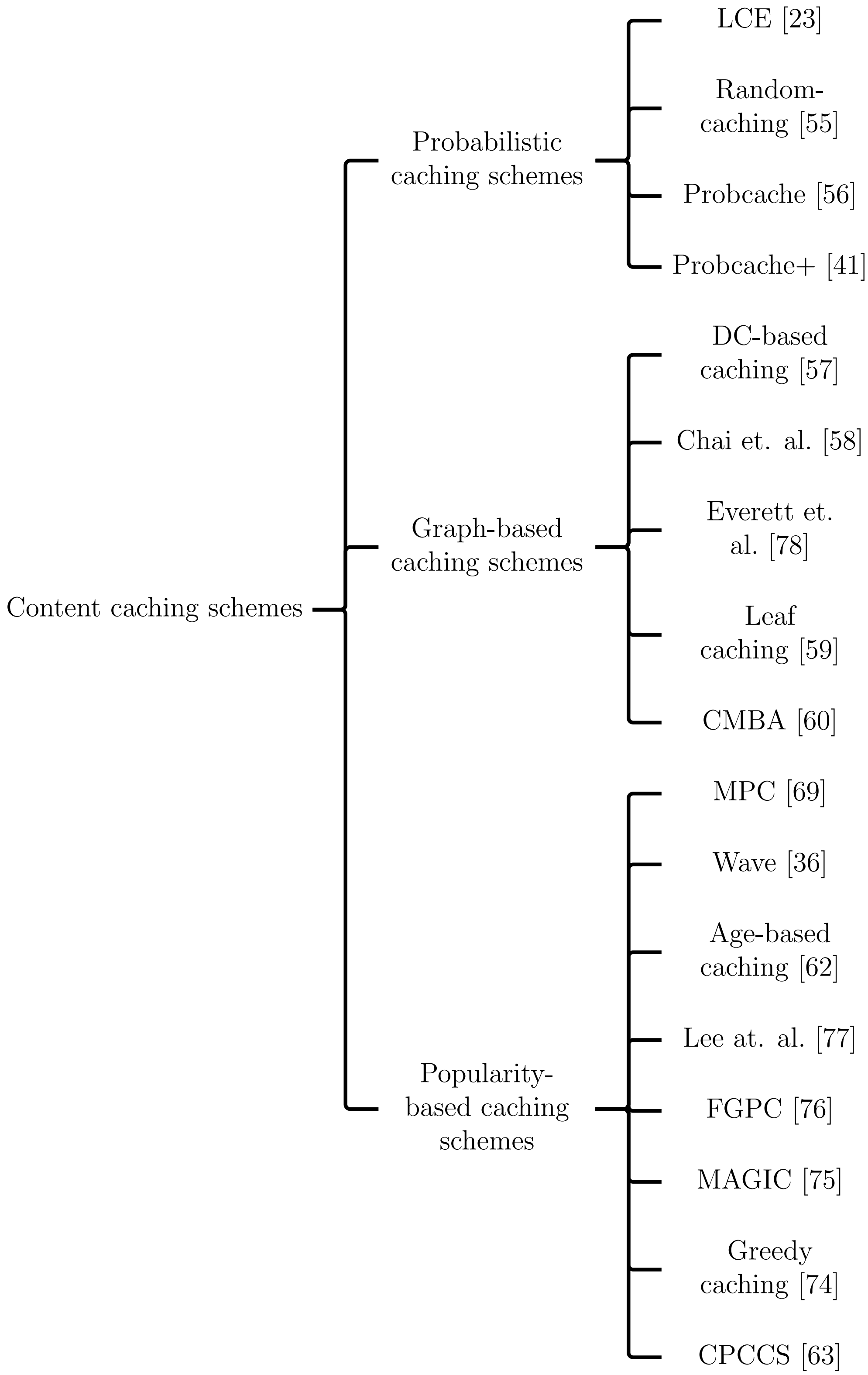


Figure 2.1: Taxonomy of content caching schemes in Content-Centric networking *(Part-1/2)*

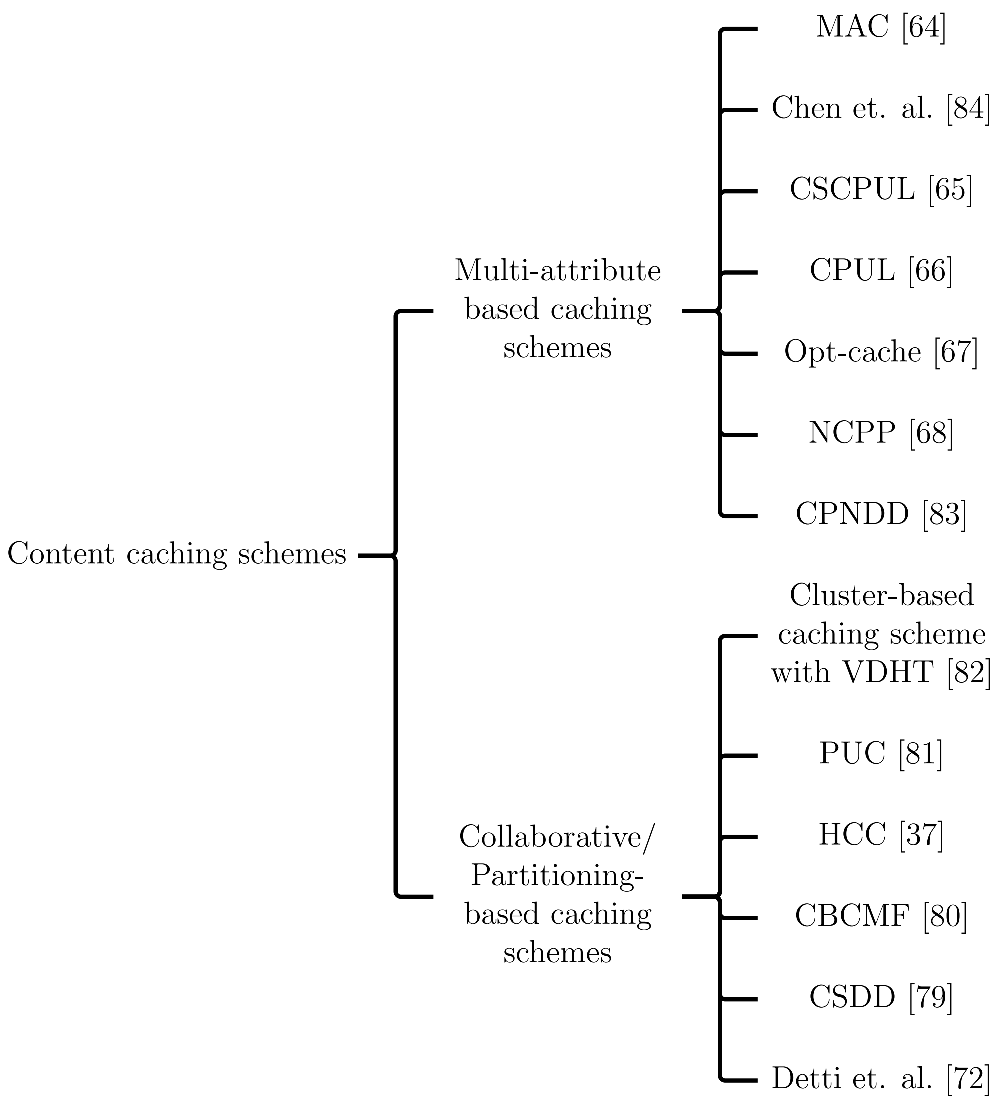


Figure 2.2: Taxonomy of content caching schemes in Content-Centric networking *(Part-2/2)*

where the requester has to generate a request to get content from the network. The scheme suggests caching the content in the intermediate routers based on a random probability. The advantage of using random caching is a low overhead of communication for placing contents and simple load sharing among nodes. Moreover, no cooperation is necessary among the network routers and each router takes the resolutions of content caching autonomously.

The *Probcache* scheme [56] proposed a caching policy for a wholly distributed and autonomous network. The Probcache strategy approximates caching space of the delivery route and probabilistically cache the contents in the intermediate routers. In the scheme, each router estimates the number of replicas of inbound contents which can be stored in the delivery route. Based on the number of cache copies and the hop count information, each router probabilistically places the content in their cache during content forwarding. Using the suggested mechanism, the scheme also leaves the cache space for the other network traffics which are sharing the same route. Hence, the scheme fairly multiplexes the contents in the network routers. During simulations, the *Probcache* strategy shows a significant reduction in the cache replacement operations as compared to the LCE caching strategy.

The *Probcache* scheme is further improved in [41], which reveals that the *Probcache* strategy suffers from biased load management due to the unfair cache weight parameter during caching decisions. The major issue with the *Probcache* is that it provides a comparatively high caching probability to those contents flows that are forwarded far from the content provider. The proposed *Probcache+* scheme recommends that the caching probability should not be directly proportional to the hop-count value from the content provider to the on-path router. Subsequently, the *Probcache+*

caching scheme shows improved load balancing and fair content multiplexing in the network.

However, both schemes [41,56] do not consider the variation in the popularity of contents during caching decisions and have no distinction in the network routers. Additionally, the execution of the suggested caching solution is not carried on the standard network topologies during simulations. To calculate the path caching capacity, each intermediate router depends on the capacity of other caches in the path. As the routers do not know caching capacity of other routers, the routers assumed that all other routers have the same cache size throughout the path. This assumption works well for those routers that have more cache space as they have more probability to store the content but it fails in fully exploiting extra caching resources. The performance of the schemes has the further scope of improvement by analyzing and considering other parameters also during content placement decisions such as the content access patterns and node centrality, etc.

2.1.2 Graph-based caching schemes

The graph-based caching schemes determine the location of the suitable router for caching decisions in the network. Using graph-based caching heuristics, the content is cached in those routers that have a higher probability of the cache hit from other routers in the network. Generally, for content placements, the routers are selected based on their location in the network or the graph-centrality metrics such as degree-centrality, betweeness centrality, eccentricity centrality, etc. These schemes improve the cache hit ratio and reduce the latency in accessing requested content because the content is placed on the routers that encounter relatively more network traffic.

The Leave Copy Down (LCD) caching strategy [61] is a simple and effective graph-based caching mechanism for CCN. On arrival of an Interest message, if the matching content is found in the cache, then the requested content is placed in the immediate downstream router of the content provider towards the requester. With an increase in the content access frequency in the network, the content is gradually placed near the requesters. However, the scheme suffers from high content redundancy and QoS degradation for the requesters as the content is placed near the requesters very slowly. A variant of the LCD scheme is called Move Copy Down (MCD). In this scheme, the content is cached in the immediate downstream router after the cache hit, and then it is removed from the storage of the content provider. Using the suggested variation, the content redundancy is restricted in the network, and therefore, more cache space would be available for different contents.

The content placement scheme proposed by Rossi et. al. [57] study the caching performance in the CCN with emphasis on the heterogeneous caching capacities of the network routers. The work analyzes different graph-related centrality parameters for caching decisions such as closeness, graph, betweeness, stress, and degree centrality. The performance of the algorithms is determined on five existing standard network topologies which are Abilene, Tiger2, Geant, Level3, and DTelekom. The graph-based parameters discussed in the paper are defined below:

1. Degree Centrality (DC): A router's degree centrality is equal to the total number of links (both in-degree and out-degree) connected with the specific router.
2. Stress Centrality (SC): A router's stress centrality is the total number of shortest routes among all nodes in the network where that router is common.
3. Betweeness Centrality (BC): It is the normalized form of stress cen-

trality. Betweeness centrality of a router is equal to the ratio of the total number of shortest routes between two nodes going through the router and the total number of the shortest path between those nodes.

4. Closeness Centrality (CC): It determines the distance of the router from all other nodes in the network. CC determines the centrality of a node in the whole network.

5. Graph Centrality (GC): The graph centrality of a router (R) is related to the distance of the router (R) to the farthest node in the network. The router with a large GC has the shortest distance to all other nodes in the graph.

The simulation results of the scheme [57] demonstrated that the node degree centrality (DC) is a robust cache allocation criterion and has good network performance for a wide variety of network topologies. Additionally, the scheme demonstrated that the gain leveraged by the heterogeneity of CCN caches is not very influential.

A betweeness centrality-based caching scheme is described in [58]. In this strategy, the router that has a higher value of betweeness centrality metrics in the delivery path caches the content during forwarding operations. The scheme eliminates the uncertainty of random caching scheme [55] by taking deterministic caching decisions.

Another betweeness centrality metric-based caching scheme is suggested in [78] for the dynamic networks. During performance evaluations of the scheme, the scheme [78] performs better than the LCE caching on Average network hop count and server load parameters. A leaf-based content placement scheme is proposed in [59] for the publish/subscribe systems. In publish-subscribe networks, the content is delivered to all the active subscribers when it is published. Therefore, generally, the network suffered from excessive content traffic as the subscribers may be disconnected or moved to other locations and the content is consuming the bandwidth and energy of the network links. The scheme recommends placing the incoming contents in leaf routers only. The other intermediate/core network routers simply forward the content towards requesters without caching operations. As the content is placed/replaced in the edge routers, the scheme shows the reduction in computational latency and the querying complexity. However, as the caching operation is performed in a small subset of network routers, the scheme does not comprehensively utilize the available cache capacity.

Lal et. al. [60] proposed a Centrality-Measure Based Algorithm (CMBA) which demonstrated that the network performance is improved by selecting the appropriate routers for content placement. It also explores the issue of excessive caching operations in the content delivery path and the scheme controls the redundancy in the network to avoid wastage of CS capacity. The proposed CMBA scheme jointly considered the degree-centrality, reach-centrality, betweeness centrality, and the closeness centrality parameters. During caching decisions, the scheme takes an average of the centrality parameters of each on-path router, and the content is placed in the intermediate routers that have the highest value of centrality measures.

2.1.3 Popularity-based caching schemes

The QoS delivered by the caching solutions is extremely dependent on the frequency and distribution of content requests to the in-network caches. In realistic network topologies, generally, a few contents become popular in the network and most of the other contents are one-timer objects [86]. One-timer contents are requested only once and then these contents are not requested again in the network, which causes poor utilization of the caching space. These one-time contents are generally lying from 45% to

75% in the caches [87]. Due to this, cache pollution issues also arise and the performance of the entire network is degraded.

For modeling of Interest arrival patterns, the majority of existing caching schemes consider Zipf distribution because it model the Interest generation patterns for most of the existing Internet applications [57, 75, 79, 88, 89]. Empirically, the Zipf distribution illustrates the access frequency for a specific content depending on its rank in the content catalog. In Zipf law, the normalized access rate of k^{th} rank content in the catalog of size $|D|$ is as follows:

$$f(k, \alpha, |D|) = \frac{1/k^{\alpha}}{\sum_{n=1}^{|D|} 1/n^{\alpha}} \tag{2.1}$$

Here, the exponent parameter (popularity skewness parameter), α represents the Interest generation patterns and it controls the Interest message generation frequency for the subset of contents in the catalog. With this characteristic, the Zipf model largely affects the performance of the network. In Zipf distribution, if the value of α is high then a large subset of Interest messages are concentrated on a smaller subset of the content catalog. Conversely, if the value of α is low, then the *Interest* messages would be more distributed for the content catalog and the popularity of contents would become more uniform. For example, smaller cache space is enough for satisfying the majority of requests with α (=1.0) as compared to (α= 0.7).

Therefore, considering the content popularity during content placement decisions is crucial for improved network performance. Towards this, a caching scheme is proposed in [35] that considers content popularity and catalog size during content placement decisions. The work illustrates that content popularity plays a crucial role in the effective utilization of the available caching capacities. The authors have also provided a simulator for CCN, which is written in C++ under OMNET++ framework as an open-source software.

The Most-Popular Content Caching (MPC) scheme [69] proposed a popularity-based in-network content caching strategy. The content placement scheme stores only popular contents in the network routers instead of caching every content. In the MPC algorithm, each node implements a popularity determination structure, which stores content name and their access frequency. If a content access frequency crosses a certain threshold value in a cache then it is marked as popular content and the router suggests other routers to cache this content using a suggestion parameter. On receiving the suggestion, other routers take caching decisions depending upon their local policies. After the suggestion process, the scheme readjusts the value of content popularity count with time to prevent excessive caching of the content in nearby routers in the network. Simulations of MPC strategy are performed over ccnSIM framework with OMNET++. The simulation performed over real-world topologies and shows a higher cache hit ratio and reduced redundancy as compared to the LCE scheme. However, the transmission of contents to the neighbors without request messages leads to increased network traffic.

The Wave caching scheme [36] shows that the optimal content placement is essential in effective content distribution and cache usage. The scheme explores efficiency in caching interrelated fragments of a single file called chunks. In this scheme, the number of chunks for caching in the routers is determined based on the content popularity. The intermediate routers which are closer to the content server recommend the number of chunks for caching in the downstream routers towards the requester. The chunks are increasingly placed in the network routers with an increase in the content popularity, which is computed using the number of Interest messages for specific content. Caching decisions are taken by individual routers independently. When a content request reaches to the content server, it responds with the 1st content chunk. Then, the chunk is placed in the immediate first router in the downstream path. When the second request comes for the same content, then the next two chunks are placed in the immediate first downstream router and one chunk is placed in the second downstream router, and so on. The algorithm shows efficiency in average hop count, caching efficiency, and cache replacement count of content delivery as compared to fixed-probabilistic and LCE caching algorithms.

An age-based content caching scheme is proposed in [62] that reduces network delay and source load in Information-centric networks. The scheme spreads popular content to network edges and completely utilizes the storage capacity of intermediate routers. The scheme eliminates unnecessary content replication in the network edges and the content caching in the routers have been managed by their respective ages. The contents obtain their initial age after storage in the cache and are evicted after age expiration. In this scheme, the network routers collaborate with each other to update the content age. If the content is closer to the network edges then it has a longer age. The cached content is replaced by incoming content if the content age is ended and the cache space of the router is full. The age of contents is determined using the distance from the server and the content popularity parameters. On receiving new content, each intermediate router determines whether it has free space available in the cache or not. If space is available, then the content is placed in the cache. If the cache becomes full, then the router determines whether it has expired content. If expired content exists then it is replaced by the newly arrived content otherwise it is not placed in the cache of the router. The proposal provides a mechanism to determine the age of any content, depending on its popularity. However, the proposal does not provide a mechanism for maintaining highly dynamic contents because those routers, which far from the server may take a longer time to refresh the content. Also, if the infrequently requested content is placed in the routers once, then highly popular content would not be able to replace that unpopular content if its age is not expired. Additionally, during simulations on CERNET2 topology, the scheme shows that the optimal relation between dynamic parameters (cache size, base age, and maximum age) has not been achieved.

A cache-capacity aware content placement scheme is suggested in [77] that provides caching in the selected on-path routers and performs cache-aware Interest forwarding. For caching decisions, the scheme considers caching capacity of routers and the content popularity together. During content forwarding, the popularity of content is used to compute the number of content copies that can be cached in the on-path routers. However, the scheme needs to determine the popularity of all the contents in the server during content forwarding and caching operations. This operation is not feasible in realistic network configurations due to high computational delay. It also causes degradation in QoS for the requesters and leads to server load balancing issues.

The Fine-Grained Popularity-based Caching (FGPC) [76] takes the caching decisions based on the content access frequencies. It always cache the incoming contents in the intermediate routers if the cache space is available. The scheme implements a popularity table in every network router to take caching decisions once the cache becomes full. The scheme stores the request messages in the table and when the number of content requests reaches a certain threshold, the content is cached in the router using the LRU replacement strategy.

The MAx-Gain In-network Caching (MAGIC) scheme [75] aims to propose a content popularity and hop-reduction parameters-based strategy for content placement decisions. The scheme also considers cache replacement penalty to reduce the number of content placement operations in the network. The scheme shows that the proposed method reduces bandwidth demand and the number of caching operations from LCE [23] and Probcache [56] strategies.

The Greedy caching scheme [74] identifies the contents to be cached in the on-path routers using the content popularity parameter. The objective of the scheme is to reduce the latency for the requesters during content access. The Greedy caching scheme approximates the popularity of contents at each on-path router based on the incoming requests from the directly connected requesters and the downstream routers. To resolve the issue of cycles in request forwarding and the duplications of content requests, the scheme creates a Directed Acyclic Graph (DAG) to estimate the relative content access rate. Using the Greedy caching mechanism, the popular contents are pushed towards the edges of the network. The simulation results on Icarus tool [90] demonstrates that the Greedy caching outperforms the LCE, LCD, and Random caching schemes on real-world network topologies.

The Compound Popular Content Caching Strategy (CPCCS) [63] explores the effect of content popularity-based caching decisions on efficient content dissemination in the network. The scheme selects the routers to cache for diverse popular contents. To determine content popularity, each router calculates the total number of received Interest messages for a specific content independently. Then, the scheme categorizes the contents into highly popular and least popular contents based on the predefined threshold value which is the average number of requests for all the contents in PIT. The scheme then selects the top one-fourth of least popular content according to their rank in content popularity. These contents are placed in the intermediate routers that are near to the requesters and most of the remaining routers are used to accommodate the most popular contents. Simulation results of the CPCCS scheme are obtained on the GEANT network topology, which show improved cache hit ratio and content diversity as compared to peer caching strategies.

2.1.4 Multi-attribute-based caching schemes

In recent, several caching solutions are proposed that consider multiple parameters during content placement/replacement decisions. These schemes suggest that involving more than one parameter during content caching decisions can utilize the available cache space more efficiently. Hence, these schemes take caching decisions by jointly considering more than one content or network characteristic.

In this direction, a Multi-Attribute Content Caching (MAC) strategy is proposed in [64] which is placing the contents in the intermediate routers after considering degree centrality characteristic, content's popularity, hop count, and the caching capacities of the routers. To provide weights to these parameters in the caching decision, the scheme uses the Z-Score normalization method. A high rank of a parameter shows a strong correlation between the network performance and that parameter. However, due to excessively considering many parameters during caching operations, the scheme imposes a very high computational overhead on the intermediate routers. This increased computational complexity become an obstacle for real-time content delivery to the requesters.

An improvement has been made in the MAC caching strategy [64] in the Chen et. al. [84]. The scheme reveals that the fixed content placement threshold value of MAC is a potential bottleneck in the efficient utilization of available cache space. The proposed caching scheme [84] takes flexible threshold standards during content caching decisions. During simulations, the scheme shows improved cache hit ratio and load reduction.

The Cache Scheme based on Content Popularity and User Location (CSCPUL) [65] attempts to take caching decisions by dividing the contents into popular and normal contents. The scheme then places the contents in the network routers after considering the content type and dispersion of user requests. For the determination of content request distribution, each router observes the Interest messages in fixed time duration. Using the proposed heuristics, the scheme cache the requested content on the suitable routers for serving specific requesters. However, the scheme is unable to place normal contents in the network routers once the caching capacity of these routers is exhausted (cache replacement cannot be performed by normal contents). Due to such limitations, the scheme has a further scope of improvement.

A Content Popularity and User Location (CPUL)-based caching scheme is proposed in [66]. The scheme classifies the contents into the categories of popular and normal contents after analyzing the location of routers where the Interest messages are sent. However, the popularity of the contents is determined by the centralized server. Therefore, the scheme does not consider the content popularity at the intermediate network routers. Moreover, only the server takes caching decisions in the entire network to optimize network performance. Due to this, the scheme is not suitable to be implemented in large-scale networks as centralized control may not be feasible. Moreover, the network routers may have their own greedy mechanism to cache the contents, and controlling these routers from the centralized server can become a difficult task.

The opt-Cache caching strategy [67] suggested a mechanism to make efficient utilization of available caching capacities to reduce resource exploitation with reduced delay. For this, the scheme simultaneously considers several parameters such as link bandwidth, content popularity, and delay with a set of defined constraints of cache size, availability, and cache replacement scheme. The scheme cache the popular contents towards the edges of the network based on the content distance from the provider. During performance evaluation, the authors demonstrated improved network performance of the proposed caching solution in terms of bandwidth utilization and delay parameters.

The Node-Content Pass Probability (NCPP) [68] caching scheme improves the network performance by taking node utilization ratio and the content popularity into consideration for content placement decisions. The appropriate resource allocation in the caching scheme improves cache hit probability in the network. The node utilization is determined based on the requester distribution, server distribution, network topology, and content retrieval routes. However, the performance of the NCPP caching scheme is compared with random caching scheme only and need to be compared with more sophisticated caching strategies on realistic network topologies.

One of our recent works co-authored by Tiwari et. al. [83] have revealed that the node degree centrality and hop count parameters are the critical parameters for content caching decisions. The proposed CPNDD (Content Placement based on Normalized Node Degree and Distance) scheme discusses the significance of considering node degree centrality and normalized hop count characteristics for caching of contents. The caching strategy makes comprehensive utilization of the cache capacities in the content delivery path by placing the contents in the important network routers. The importance of a node is determined using the node degree centrality characteristic. The caching probability increases as the content message traversed towards the requesters. Thus, the content requests need not to be traversed to the server and the matching contents are accessed closer to the boundaries of the network. The performance of the CPNDD solution is compared with traditional LCE, DC-based, and distance-based caching strategies. The simulation results show improved cache hit ratio and lesser network hop count for the CPNDD scheme as compared to existing schemes[1]. However, the scheme may suffer from high content redundancy in highly skewed popularity distributions due to autonomous content caching decisions. Moreover, the CPNDD caching scheme does not consider content popularity during the caching decision, which can largely affect the performance of the network.

Consequently, a careful investigation has been made on the CPNDD scheme in a heterogeneous network having different connection bandwidths. The review of the scheme revealed that the performance of the CPNDD caching solution can be further enhanced by incorporating the bandwidth parameter along with the node degree centrality and hop count parameters in the content placement decisions. Towards this, an improved caching scheme is proposed in one of our prior works [91] that computes the distance between two network routers based on the minimum bandwidth and the hop count value for the Interest message to reach the content provider and the bandwidth of the links traveled by matching Content message. Using the suggested caching mechanism, the content placement probability improves for important in-network routers which are farther from the content provider and are connected using a lesser bandwidth connection in the delivery path. The scheme decreases the network congestions due to increased caching probability closer to the boundaries of the network.

[1] The major findings of proposed CPNDD scheme have been published

- "An efficient content placement scheme based on normalized node degree in content centric networking". Cluster Computing, 24(2), 1277-1291, 2021. (SCIE)

Simulation results show that the proposed caching solution outperforms the existing caching schemes[2].

However, the suggested mechanism also takes autonomous caching decisions and therefore, can suffer from high content duplication in the network. Additionally, the caching performance may degrade if on-path routers have identical degree centrality metrics and are connected using homogeneous connection bandwidths.

2.1.5 Collaboration and/or Partitioning-based content caching schemes

Different research works have shown that the network bandwidth usage and latency are reduced up to a certain extent in those on-path caching schemes where intermediate routers take autonomous caching decisions [92]. Although on-path caching strategies improve QoS for the requesters, there is a further scope of betterment in the cache space utilization. The autonomous content placement schemes suffer from high content duplication in the network and limits cache diversity. To address this issue, many collaboration or network partitioning (clustering) based in-network caching schemes are proposed recently to reduce excessive redundancy of contents and improving network performance. In the partitioning-based algorithms [73, 82, 93] each partition can cache at most one copy of content during content forwarding while partitions may cache the same copy of the content. With partitioning of the network, high cache diversity is maintained in the intra-partition communication.

A dominating-set-based collaborative content placement approach is

[2]The major findings of the scheme are published

- "Minimizing delay in content-centric networks using heuristics-based in-network caching". Cluster Computing, 1-15, September 2021. (SCIE) proposed in [85]. The scheme takes the arbitrary topology and creates a virtual backbone network. The backbone routers are designated as core routers and the remaining routers become the regular routers. Then, the scheme recommends a collaborative caching strategy that jointly considers the content placements and the routing information. The content caching is allowed in the core routers only and the regular routers do not cache the forwarded contents. The scheme uses an LFU cache replacement scheme when caching capacity is completely exhausted. Once the content is evicted from the core routers then it is randomly sent to one of their adjacent regular router for caching. Although the scheme shows improved cache hit ratio and delay from peer caching schemes, the suggested mechanism causes excessive network traffic and congestions as the evicted contents are transmitted to the connected regular routers without any request for them. It also increases link load in the network and poor utilization of the available network bandwidth.

The cluster-based caching with VDHT (Virtual Distributed Hash Table) scheme [82] proposed a cluster-based in-network caching mechanism for reduction of content duplication and increase in cache hit-ratio. The scheme suggests the K-Medoid clustering algorithm to cluster the network into k clusters. The VDHT is used to efficiently manage the resources within the clusters. The scheme also proposed inter-cluster and intra-cluster routing algorithms for cluster-based in-network caching. During the execution of the clustering mechanism, the initial cluster heads are selected based on the node degree centrality parameter. During the clustering process, the distance between a router and the cluster heads is computed based on the delay, bandwidth, cache size, and hop-count parameters. The intra-cluster routers use a hash function to determine the location of specific content. VDHT combines the cache state of all routers within a cluster using the hash function. When an Interest message arrives for content, its location is computed using the hash value of the content identifier and it is forwarded accordingly. The experimental results of the scheme show that the cluster-based algorithm outperforms LCE [23] and Probcache [56] strategies in terms of the cache hit ratio and link load metrics. However, the scheme does not consider the content access frequencies during content placements which is one of the critical parameters that affect the network performance. In addition, this scheme requires knowledge of the entire network topology as the number of clusters needs to be fixed initially, which is one of the major issue in large-scale networks.

The k-split [94] and k-medoids [95] based content placement schemes are proposed in Sourlas et. al. [96]. The k-split scheme divides the network topology into *k-partitions* to decrease inter-partition distance based on the similarity parameter i.e. latency. Similarly, the k-medoid partitioning scheme selects the *k-routers* that are designated as the initial centers of each partition. The betweeness centrality measure is considered to select first *k-partition centers* to largely influence the forwarding of contents within partitions. For content caching decisions, the scheme uses the bin packing content assignment function with a hash function. However, determining the optimal number of partitions becomes a big concern in realistic network topologies. Moreover, the scheme requires knowledge of the entire network during network partitioning. The scheme also increases the communication overhead as the hash-based off-path caching scheme is used for content caching operations.

The Packet Update Caching (PUC) strategy [81] addresses the issues related to the hop count reduction, access control, and network traffic. The scheme is based on partitioning of the network and the partition heads aggregate the requests from the interrelated routers. These aggregated requests are then forwarded towards the content provider. The partition heads are selected after considering the exhausted energy, degree centrality and the distance parameters [97]. For content retrieval and caching operations, the scheme uses a circular buffer that has two pointers for reading and writes operations. The scheme proposes that if the content requester and the provider router exist in the same partition, then no caching operation is performed to minimize content redundancy within partitions. When the content is not found within the partition then it is retrieved from the server and a replica of the requested content is placed in the partition head of the requester.

The Hierarchical clustering (HCC) scheme [37] addresses various challenges that exist in the content-centric paradigms such as limited cache capacity and communication overhead during cache management. It proposes a two-layer hierarchical cluster (partitions) based solution to enhance in-network caching efficiency. The whole network is divided into several clusters and the cluster heads are selected in each partition for caching decisions. The content popularity and the location of routers are jointly considered during content placement operations. The technique is suitable for distributed networks such as Mobile Ad-hoc Networks (MANET), Wireless Sensor Networks (WSN), and Vehicular Ad Hoc Networks (VANET). The design proposes two-level hierarchical clustering in which the core layer routers are not used for caching and the edge layer cache the incoming contents. The edge layer can be divided into many edge clusters depending upon the number of routers in the edge layer. The cluster heads are elected by the cluster members in a decentralized manner based on the node degree, average transmission time, and weighted hop count parameters. The network designates the gateway routers to connect two different clusters. Member routers are the remaining routers that are not designated as the cluster head or gateway. These members cache the content based on the content popularity and location of routers. In this scheme, only the cluster head can flood a request to all routers within the cluster and other routers communicate with the cluster head to request the content. The simulation results show improved network performance on hop reduction and average latency reduction parameters as compared to Probcache, LCE, and LCD schemes.

A Cluster-Based Content Management Framework (CBCMF) is proposed in [80]. The proposal provides a partitioning-based content administration framework for ICN with content registration, dispersal, retrieval, and caching. The scheme uses an algorithm discussed in [98] for cluster formations. It creates and maintains a small number of clusters (cliques) in the network topology. The cluster head is directly connected with all routers within the cluster and stores all incoming/outgoing content of cluster members and information of neighboring cluster heads. The request messages from the end-user devices are received by the adjacent routers which transmit them towards the cluster head. Cluster head checks its PIT for matching content requests. If no match is found then it creates an entry in its table and forwards the request to adjacent cluster heads. Consequently, the content delivery takes the symmetric path towards the requester. In the framework, a requester can also check the validity of the content name before registering it within a content provider. The in-network content placement decisions are taken after considering the content popularity and time-dependency nature of the content. The simulations results show that the suggested caching strategy decreases the content transfer times.

In recent, the CSDD (Collaborative Caching Strategy Distance and Node Degree) caching strategy [79] suggests collaboration among network routers and takes caching decisions after analyzing the node degree centrality and the hop count parameters. During content forwarding, each intermediate router first computes the percentage of the route traversed by the content message during content delivery. If the route covered by the content is not greater than the predefined threshold value, then the content is not placed in the router. If it is greater than the threshold, then the degree centrality of the router is taken into consideration during caching decisions. Using the recommended heuristics, the contents are placed in the important routers and ensure content availability in the network. During simulations, the scheme shows improved network lifetime and path stretch metrics on the traditional LCE [23] and LCD [61] caching strategies.

Detti et. al. [72] proposed a partitioning-based caching scheme to distribute the load of the router on various physical devices. The scheme address the design challenges in developing the partition-based router. In this caching strategy, the overall request forwarding rate is increased/decreased by appropriately maintaining the number of partitions and based on the caching performance of each partition in the network. The internal architecture of the partitions consists of several collaborative routers that are seen as a single router from outside of the partition. Therefore, internally, the partition works as a load-balancer among its routers. However, for load-balancing, the CSR (Cluster-based Scalable Router) evenly distributes the incoming requests in all internal routers based on a hash function. Therefore, although the scheme [72] improves the distribution of content requests, it fails to achieve optimal network performance as the scheme does not consider the router's and content's characteristics during caching operations.

2.2 Gaps in Literature

1. Consideration of content popularity characteristic has not been taken by many caching schemes. During content placement decisions, the weightage of the content popularity can play a major role in improving content retrieval performance.
2. Most in-network caching algorithms use broadcasting of Interest messages to locate the content. It creates higher network traffic and makes the network congested. This subsequently increases computational overheads on the intermediate routers for maintaining their FIB and PITs. Many research proposals do not consider the broadcasting overhead.
3. A single parameter-based content caching scheme is unable to leverage the available caching capacity efficiently. Hence, there is a need to consider more than one network topology-based and content-based parameters.
4. Many collaborative/partitioning-based caching schemes require knowledge of the entire network to take efficient content placement decisions. This might not be feasible in large-scale networks. It also increases communication and computation overhead on the network routers.
5. The off-path caching schemes consider the cache status of all Content Stores during cache placement decisions. It may increase their metadata size, due to which the efficiency of the scheme can degrade for highly dense networks. Therefore, proposals on off-path caching are not able to offer a viable solution for cache placement of required QoS parameters. Therefore, there is a need for some efficient on-path caching scheme for content caching and its faster retrievals.

6. Many partition-based schemes exploit k-medoid and k-split algorithms for network partitioning. These schemes are static and create a fixed number of partitions in the network, while dynamic partitioning is desirable for different types of network topologies. Therefore, network-dependent partitioning is required.

2.3 Analysis

This section discusses the challenges behind the proposed work of the thesis. Various content placement and replacement schemes are reviewed in this chapter which indicates the issues with the router selection, content selection, and the need for efficient collaboration among network routers during caching decisions.

The chapter also describes the issues that exist in traditional content caching schemes, namely, cache hit ratio, delay in retrieving the content, number of hops traversed to retrieve the content, and network traffic overheads. These performance metrics have a significant effect on the network performance and QoS for the requesters.

2.4 Summary

This chapter explored the existing content caching schemes which are categorized into probabilistic, graph-based, popularity-based, multi-attribute-based and collaborative/partitioning-based strategies. The chapter discusses the advantages and disadvantages of taking autonomous and collaborative content caching decisions. The analysis of the existing caching schemes has been carried out using the network nodes characteristics, content metrics, and partitioning needs. After the review of recent caching schemes, it has been concluded that efficient utilization of the available caching space is required in CCN. In the next chapter, a content caching scheme is proposed for the CCN.

Chapter 3

Proposed Content Popularity and Distance-based Caching Scheme (PDC)

This chapter proposed an on-path content caching scheme named PDC (Popularity and Distance-based Caching Scheme) [99] that takes content placement decisions based on content popularity and distance-based parameters. By considering these parameters, the scheme places popular contents near the requesters. Therefore, the scheme improves the QoS for the requesters (consumers) and decreases network traffic[1].

3.1 Problem Statement

In real-world content access patterns, the requesters always access many of the unpopular contents that are not requested frequently in the net-

[1]The major findings of this chapter have been published

- "Optimized content centric networking for future internet: dynamic popularity window based caching scheme". Computer Networks, 179, 107434, 2020. (SCIE)
- "Dynamic popularity window and distance-based efficient caching for fast content delivery applications in CCN," Engineering Science and Technology, an International Journal, 24(3), 829-837, 2021. (SCIE)

work. Placing the infrequent contents in the routers leads to poor usage of the cache space. Moreover, many of the existing popularity-based caching schemes determine the popularity of contents by analyzing the Interest messages that are received in previous “n seconds”. However, it is apparent that different routers have dissimilar characteristics in the network and receives a varying number of content requests in the same duration. Moreover, much of the existing research work considers fixed-threshold values for the identification of the popular contents in the network. Determination of popular contents using time-based duration and fixed popularity threshold might not detect the frequently accessed contents reliably in the network routers. Hence, generally, the caching schemes with these heuristics have caused degraded network performance and QoS for the requesters. The existing probabilistic and graph-based caching strategies suggest that the network performance is improved by caching the incoming contents near the requesters. When the content is placed near the requesters, the future requests for the cached contents are served from the nearby routers instead of forwarding every Interest message to the server. Therefore, for improving the cache hit ratio and reducing the network delays and traffic, the need for an efficient caching scheme has appeared that makes caching decisions after analyzing the content popularity and distance-based characteristics.

To determine the content popularity with reliability, the scheme proposed a novel dynamic-sized content Popularity Window that analyzes the incoming Interest messages and identify frequently accessed contents. The proposed strategy should optimize content placements in the network routers using the novel Popularity Window and distance-based parameters. This optimization problem of content caching is defined in the below-mentioned equation 3.1:

$$Minimize\{Fn(\tau, H(I_j), H(D_j), \mu, \delta)\}$$

$SubjectTo$: Caching Decisons on $T_{R_i}^{D_j}$

$$\begin{cases} If\ T_{R_i}^{D_j} \leq W_{R_i}(C\_name(D_j)) : Content\ Caching \\ Otherwise : No\ Caching\ operation \\ Where : \\ 1.1 \geq \alpha \geq 0.7 \\ C_{size}(R_i) << |D| \\ 0 \leq \delta \leq 0.6 \end{cases} \tag{3.1}$$

The notations presented in the equation are summarized in the Table. 3.1.

As defined in equation 3.1, there is a need to design an efficient in-network content caching scheme for effective exploitation of available cache space and to minimize network traffic and content retrieval delay.

3.2 System Model

3.2.1 Network Model

For convenient discussion of the system model for proposed PDC caching scheme, the notations and their definitions have been illustrated in Table 3.1. Suppose, $G(V, E)$ be a network configuration as illustrated in Fig. 3.1. In this topology, the symbol V corresponds to the set of nodes in the network; $V = \{U_1, U_2, ..., U_{|U|}, R_1, R_2, ..., R_{|R|}, Server\}$. The network model of the proposed PDC strategy consists of 3 types of network nodes called users/requesters (U_i), network routers (R_i), and the $Server$. The links among these nodes are represented as the members of set E and these links are used to forward Interest/Content messages and control information among network nodes. As shown in the example of network topology, the server is connected with all network routers and requesters through single-hop and multi-hop links. The server holds the entire content catalog

Table 3.1: Variables definitions

Variable	Definition
$\|D\|$	Content catalog size.
$\|R\|$	Total number of in-network routers in the network.
$C_{size}(R_i)$	Content Store (CS) capacity in R_i.
$CS(R_j)$	Content store of router R_i.
U_k	k^{th} requester in the network.
$W^s_{R_i}$	Popularity Window capacity in R_i.
$Max(W^s_{R_i})$	Maximum capacity of Popularity Window in R_i.
$S^l_{R_i}$	l^{th} slot of Popularity Window in R_i.
I_j	j^{th} Interest message.
$C\_name(I_j)$	Content name in I_j.
$C\_name(D_j)$	Content name in Content message D_j.
$W_{R_i}(C\_name(D_j))$	Count of $C\_name(D_j)$ entries in W_{R_i}.
θ	Coefficient for Popularity Window capacity.
λ	Request generation rate by every requester in per unit time.
τ	Network Delay in retrieving the requested content.
μ	Network Traffic in KB/s.
$AvgPop(W^s_{R_i})$	Average popularity of Interest messages in the Popularity Window.
$Norm\,(H(D_j))$	Normalized hop-count value traversed by D_j.
$H(I_j)$	Hop-count taken by I_j to reach the content provider.
$H(D_j)$	Hop-count traversed by D_j from the content provider.
$T^{D_j}_{R_i}$	Threshold value for caching the content D_j in R_i.
α	Exponent value in Zipf distribution.
$Dis(W^s_{R_i})$	Number of distinct Interest messages in $W^s_{R_i}$.
δ	Coefficient for normalized hop-count in threshold computation.
$Dist(I_k)$	Distance traversed by the *Interest* message I_k.
$Dist(D_k)$	Distance traversed by the *Content* message D_k.
P_x	x^{th} partition in the network.
$Dist(P_x, P_y)$	Distance between partition P_x and P_y.
B	Average connections bandwidth among network routers.
$H(P_x, P_y)$	Number of hops in the shortest path between the farthest routers belonging to partitions P_x and P_y.
$b\_width(P_x, P_y)$	Minimum bandwidth in the shortest path between farthest routers of partitions P_x and P_y.
$S.PH(D_i)$	Partition head name of source/content provider.
$O.PH(D_i)$	Partition head name of the on-path router that have forwarded D_i.
$H^{I_i}_{R_j}$	Cache hit or miss for the Interest message I_i in R_j.
$\phi(D_i)$	Control the caching decisions for D_i.

which can be requested in the network and hence, the server act as a sink for all the Interest messages with cache hit probability=1.

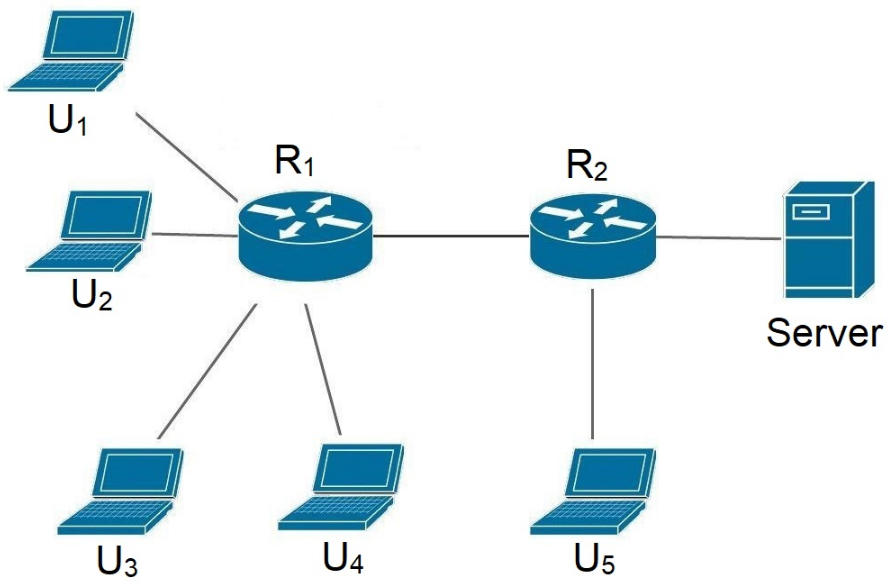


Figure 3.1: An illustration of network topology in CCN

The set of different contents is represent with $Content\ catalog\ D = \{D_1, D_2, ..., D_{|D|}\}$, where each content is represented as D_j; $1 \leq j \leq |D|$. When requester (U_k) need to access a content D_j, then it prepares the corresponding Interest message I_j, where $C\_name(I_j)$ is identical to $C\_name(D_j)$. The Interest message I_j is forwarded to its neighboring router which connects the requester with the network. The network routers execute Interest message and Content message forwarding procedures and have content caching capability. The content is cached by the routers based on the applied caching heuristics. In general, the Content message follows a fixed packet architecture, and therefore, it is fairly rational and assumed that all the Content messages have uniform packet size in the proposed caching scheme. It is presumed that the cache storage size of the routers is very small than the content catalog size as defined in many realistic network topologies and it has been represented as follows:

$$C_{size}(R_i) << |D| \tag{3.2}$$

3.2.2 Dynamic model and assumptions

In CCN, the Interest message needs to be routed in the network to retrieve the requested content. When a user forwards an Interest message I_j, then it has been assumed that the Interest message uses the *best-route forwarding strategy* [100] in the network to find the matching content with minimal delay. In CCN, the Interest message is forwarded link-by-link using this strategy until the Interest message reaches the content server or any intermediate router is found which is having a copy of matching content. The best-route routing strategy is also deployed for the existing competing caching strategies during performance evaluation.

The proposed scheme assumed that the content access pattern follows Zipf distribution [101, 102] which is discussed in the previous section. In Zipf distribution, the popularity skewness parameter α (also called exponent) controls the content access distribution from the catalog. Fig. 3.2 shows the rank of 80^{th} percentile content in the Zipf distribution for $|D| \in \{500, 5000, 50000\}$ and $\alpha \in \{0.1 - 1.5\}$. As illustrated in Fig. 3.2, when the value of α increases, most of the Interest messages are concentrated for a smaller subset of the content catalog. The graph also shows that when $0 < \alpha < 0.4$, the requests are more distributed for different contents in the network, and hence, no content is significantly popular than the other contents. However, in the majority of realistic networks, few contents are more popular than the remaining contents, and therefore, a comparatively smaller portion of the content catalog is required to be cached for serving 80% of the Interest messages from available caching space. Existing research shows that the value of $\alpha \in \{0.7 - 1.1\}$ is more suitable to model

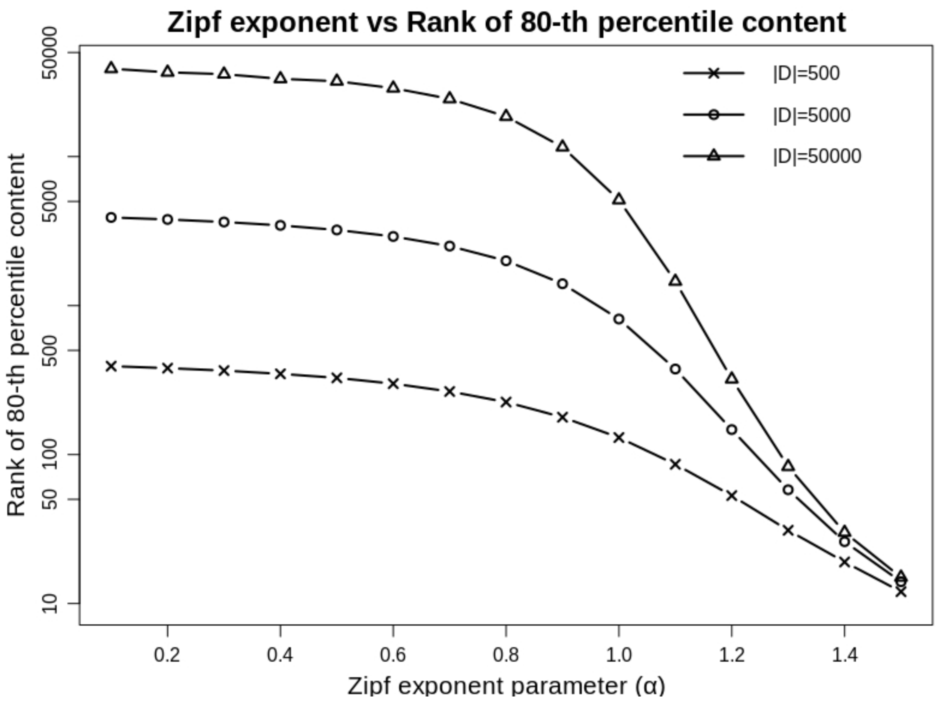


Figure 3.2: Rank of 80-th percentile popular content for different catalog sizes according to Zipf distribution vs value of Zipf exponent parameter

the content access patterns in real network configurations [37, 47, 56, 58]. Hence, the Zipf distribution is considered with $\alpha \in \{0.7 - 1.1\}$ to model content access patterns.

To identify the popular contents during content message forwarding, the PDC strategy implements a new structure called "Popularity Window" [103] in each in-network router along with PIT, FIB, and CS tables. The Popularity Window is dynamically sized based on the content catalog size and has slots to stores the requested content names ($C\_name(I_j)$) from the Interest messages. Without losing generality, it is assumed that each slot of the Popularity Window can store one content name. When an Interest message arrives, the router stores the name of requested content in the slot of Popularity Window. It is also presumed that the network nodes are stationary and do not have issues related to energy constraints.

3.3 Design and operations of Dynamic Popularity Window

As defined, the Popularity Window is implemented in every in-network router to determine the content's popularity. Instead of fixing the size of Popularity Window arbitrarily, it has been observed that its size should be directly proportional to the content catalog size. If the content catalog is too large, then the content popularity cannot be determined reliably using a smaller Popularity Window due to the subtle difference between the access frequency of popular and infrequent contents. Therefore,

$$Max(W^s_{R_i}) \propto |D| \tag{3.3}$$

Hence, to improve the network performance and determining the popular contents accurately, the size of the Popularity Window is computed based on Equation 3.4 as follows.

$$Max(W^s_{R_i}) = \lfloor \theta \times |D| \rfloor \tag{3.4}$$

In this Equation, θ is a regulatory coefficient and the size of the Popularity Window is proportional to θ. The optimal value of θ is determined based on the computational efficiency of the in-network routers. It works as a tuning nob in the network because if θ is too large then the content popularity would be computed in a large Popularity Window with improved accuracy. Contrarily, if θ is too small then the determination of content access frequency would be less accurate but it would decrease the computational duration and provide faster content delivery. Hence, the value of θ can be adjusted based on the computational efficiency of network nodes and desired level of reliability in the identification of popular contents.

The mechanism of insertion and eviction of the names of the requested

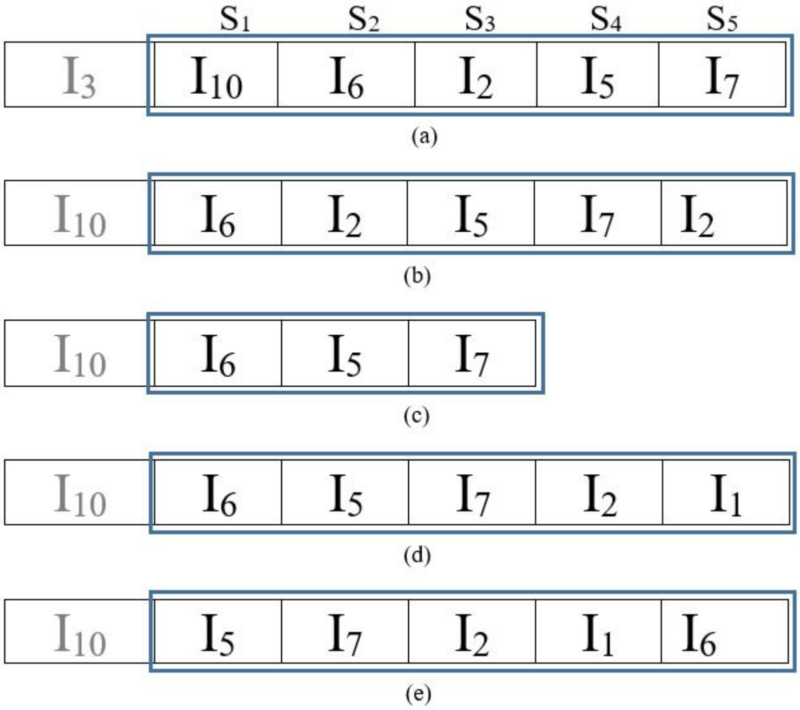


Figure 3.3: Functionality of Popularity Window in each in-network router

contents from the Popularity Window are depicted in Fig. 3.3. As shown in $step-(a)$ in Fig. 3.3, the maximum size of the Popularity Window is 5 and it is having the content names of Interest messages $\{I_{10}, I_6, I_2, I_5, I_7\}$. When a new Interest message (I_2) arrives and no slot is free in the Popularity window ($Max(W^s_{R_i}) = 5$), then the information of the oldest Interest message is removed from the first slot and the remaining requests are left-shifted in the Window ($step - b$). Then, the requested content name ($C\_name(I_2)$) is stored in the Popularity Window. When a matching content is received for Interest message I_2 and it is cached in the router, then its entries are removed from the Popularity Window as shown in $step-(c)$ in Fig. 3.3. In $step-(d)$, the information of two incoming Interest messages I_2 and I_1 are stored in the empty slots of the Popularity Window as ($Max(W^s_{R_i}) = 5$). When content is received and it is not cached in the router (I_6) (in the example shown in $step - e$ of Fig. 3.3), then the information of its oldest Interest message is moved to the most recent slot of the Popularity Window to improve its caching probability during future content requests.

The existing research works [41, 56] shown that the network performance is improved with an increase in the distance between the content provider and the on-path routers. Therefore, the proposed caching scheme decreases the threshold value with increase in hop count value to improve the caching probability. Here, the proposed caching scheme jointly considers the content access pattern and the distance traversed by the Content message during caching decisions as shown in Equation 3.5.

$$T_{R_i}^{D_j} = \left[AvgPop(W_{R_i}^s) \times [1 - \delta \times Norm\,(H(D_j))]\right] \tag{3.5}$$

As shown in Equation 3.5, the value of $T_{R_i}^{D_j}$ decreases with an increase in the number of hops traversed by the Content message (D_j).

The value of the $AvgPop(W_{R_i}^s)$ and $Norm\,(H(D_j))$ parameters are computed using following equations:

$$AvgPop(W_{R_i}^s) = \frac{W_{R_i}^s}{Dis\left(W_{R_i}^s\right)} \tag{3.6}$$

$$Norm\,(H(D_j)) = \frac{H(D_j)}{H(I_j)} \tag{3.7}$$

Using Equation. 3.6 and 3.7, Equation. 3.5 can be rewritten as follows:

$$T_{R_i}^{D_j} = \left[\frac{W_{R_i}^s}{Dis\left(W_{R_i}^s\right)} \times \left(1 - \delta \times \frac{H(D_j)}{H(I_j)}\right)\right] \tag{3.8}$$

3.4 Interest message processing

To access a content (D_j), the requester first needs to prepare an Interest message (I_j) and initialize a field $H(I_j)$ to 0. The requester then forwards I_j to its adjacent router in the network. After receiving I_j on the incoming face, each intermediate router R_i executes the "Interest message processing procedure" as illustrated below.

Interest message processing procedure $(I_j,\ R_i)$

1. $H(I_j) = H(I_j) + 1.$
2. *Search* $C\_name(I_j)$ *in the CS of* R_i.
3. *If the content exists, then follow Content message processing procedure mentioned in section 3.4.1.*
4. *If* $C\_name(I_j)$ *not found in the CS, then search PIT. If entry found for* $C\_name(I_j)$*, then aggregate Interest message in PIT and remove* I_j *from the network.*
5. *Otherwise,* R_i *searches its FIB and based on the information from FIB, the router forwards the Interest message in the network.*
6. The below-mentioned steps store the requested content name in the Popularity Window from incoming message I_j.
   (a) *If* $S^1_{R_i} = NULL$ *then,*

   Initialize $k = 1$ *for router* R_i

   Move $S^k_{R_i} \leftarrow C\_Name(I_j)$

   (b) *If* $Max(W^s_{R_i}) > (W^s_{R_i})$ *then,*

   set k=k+1 for router R_i

   Place $C\_Name(I_j)$ *in slot* $S^k_{R_i}$

   (c) *If* $Max(W^s_{R_i}) = (W^s_{R_i})$ *then*

   Remove I_j *from* $S^1_{R_i}$

   do l=2 to $Max(W^s_{R_i})$

   Move $S^{(l-1)}_{R_i} \leftarrow S^l_{R_i}$

   Move $S^l_{R_i} \leftarrow C\_Name(I_j)$

The router R_i first increases the number of hops traversed by I_j in the field $(H(I_j))$ as mentioned in $step - 1$ of the procedure. Then, R_i

matches the stored contents with the requested content as defined in $step-2$. If matching content D_j is found, then R_i follows the "Content message processing procedure" as mentioned in section 3.4.1. $Step-2$ to $step-5$ discuss the traditional Interest message processing of CCN [23, 34, 104].

In $step-6$, the name of requested content $(C\_name(I_j))$ is stored in the Content Popularity Window which is used during the caching decisions. When Popularity Window is empty and the first Interest message arrived, then its information is stored in the first slot of the Window as shown in $step-6(a)$. Otherwise, the incoming Interest message name is stored in the empty slot of the Popularity Window. Once the maximum capacity of the window reaches and incoming $(C\_name(I_j))$ need to be stored, then the existing Interest names are left-shifted in the Window by 1 position, and the $(C\_name(I_j))$ is stored in the most-recent slot (illustrated in $step-6(c)$).

3.4.1 Content message processing

Suppose, a matching content is found for Interest message (I_j) in the cache of router R_i or request is served by the server (called as $Content\_Provider$) then it perform $step-1$ of the *Content message processing precedure*. The $Content\_Provider$ prepares a matching Content message (D_j) with name $C\_name(D_j)$ and replicate the value of $H(I_j)$ field from I_j to corresponding field in D_j. The provider set the value of field $H(D_j)$ to 0 and forward the message towards the requester $(step-1(b)-(c))$. When D_j is received by an intermediate router R_k, then it performs $step-2$ to increase the number of hops traversed by D_j. The R_k then execute $step-3$ or $step-4$ based on the matched condition. $Step-3$ is executed when D_j is popular $\left(W_{R_k}(C\_Name(D_j)) \geq T_{R_k}^{D_j}\right)$ where, $T_{R_k}^{D_j}$ is computed using Equation 3.8. In this scenario, the D_j is cached in the CS of R_k using LRU cache replacement policy $(step-3(a))$. After caching, R_k removes all the requests of this Content message from the Popularity Window as shown in $step-3(b)$ and set the value of $H(D_j)$ field to 0 to avoid excessive caching operations for D_j. Otherwise, if the content is not popular enough or it has not traversed sufficient distance $\left(0 \leq W_{R_k}(C\_Name(D_j)) < T_{R_k}^{D_j}\right)$, then $step-4$ is executed. In $step-4(a)$ and $step-4(b)$, the oldest entry of the Popularity Window which is matching with the incoming message D_j is shifted to the last slot of Popularity Window to increase its caching probability during subsequent content retrievals. After caching decision, the content is forwarded towards the requester ($step-5$).

Content message processing procedure (D_j, R_i, R_k)

1. *If R_i is Content_Provider then*
   (a) *Create matching D_j with payload.*
   (b) *Copy the value of $H(I_j)$ from I_j to D_j.*
   (c) *Set the value of field $H(D_j) = 0$.*
2. *Else,* $H(D_j) = H(D_j) + 1$
3. If $W_{R_k}(C\_Name(D_j)) \geq T_{R_k}^{D_j}$ then
   (a) *Cache content in the CS of R_i using LRU replacement scheme.*
   (b) *for $1 \leq l \leq (W_{R_k}^s)$*
   Remove $C\_Name(D_j)$ from $S_{R_k}^l$;
   Set l← (l-1)
   (c) $H(D_j) = 0$
4. *Else If $0 \leq W_{R_k}(C\_Name(D_j)) < T_{R_k}^{D_j}$ then*
   (a) *do l= pos($C\_Name(D_j)$) to $Max(W_{R_k}^s)$*
   Move $S_{R_k}^{l-1} \leftarrow S_{R_k}^l$
   (b) *Move $S_{R_k}^l \leftarrow C\_Name(D_j)$*
5. *Forward content towards requesters according to PIT entries of R_k*

3.5 Cost analysis of the Dynamic Popularity Window in PDC strategy

In PDC caching solution, each in-network router implemented a Dynamic Popularity Window in its storage to determine the content popularity. The Popularity Window size is intelligently computed after considering the content catalog size as defined in Equation 3.4 to minimize the loss of cache space. In the standard packet formats of Interest and messages, the contents are uniquely identified using a 32-bytes field for their names $(C\_name(I_j))/(C\_name(D_j))$. For instance, in order to store 10^6 content names, the Popularity Window creates an overhead of $32 \times 10^6 Bytes$ or $32\ MB$ on the cache which is rational to the improved network performance achieved by the proposed caching scheme.

To examine the computational overhead of implementing the Dynamic Popularity Window, the $step - 6(a) - 6(c)$ of the "Interest message processing procedure" and $step - (3 - 4)$ of the "Content message processing procedure" are analyzed. These steps involve the computations performed in the Popularity Window for determining the popularity of contents. As illustrated in $step - 6(a) - 6(c)$ of "Interest message processing procedure (section 3.4)", the worst-case asymptotic time complexity is $O(Max(W^s_{R_i}))$ for storing the Interest message names in the Popularity Window. Likewise, the worst-case asymptotic time complexity for content popularity computation from the Popularity Window is $O(Max(W^s_{R_i}))$ during execution of $step-(3-4)$ of the "Content message processing procedure (section 3.4.1)". Hence, there is a linear relationship between Popularity Window size and the time complexity to perform operations on it. As nowadays, the network devices have powerful processing characteristics, therefore the computational and space overhead of the Popularity Window is subtle and the proposed PDC caching scheme can be readily implemented in realistic network configurations.

3.6 Limitations of proposed PDC caching scheme

The proposed PDC strategy has been initially proposed and published in [99]. But, a careful investigation of the scheme reveals that sometimes, the autonomous content caching approach of the proposed caching scheme increases content redundancy in on-path routers. This occurs when the number of requests for the popular contents are very high and these contents are placed in the network routers in an unrestrained manner. This increases the server load and network traffic as content duplication is very high in the network routers and the remaining content requests need to be served by the server. Therefore, there is a need to propose an efficient network partitioning mechanism where intra-partition routers collaborate with each other to reduce the content redundancy and load from the content server. Consequently, the QoS for the requesters would be further improved as contents will be retrieved with lesser delay.

3.7 Summary

This chapter presents the proposed content caching scheme called PDC, which explores the significance of content popularity and distance-based parameters in cache space utilization. For content popularity computations, the scheme implements a novel Content Popularity Window structure in each network router. The chapter also summarizes the limitations of the PDC strategy. In the next chapter, a network partitioning-based caching scheme is proposed for the efficient utilization of available caching resources by controlling content caching operations in the network partitions.

Chapter 4

Dynamic Partitioning and Popularity based Caching for Optimized Performance (DPPCOP)

This chapter discusses the proposed network partitioning scheme and heuristics to determine suitable partition-heads in the network topologies. The chapter also elaborated the novel Interest and Content message forwarding and the caching mechanisms for improved network performance.

4.1 Problem Statement

As per the regress literature review and the analysis of the limitations in the proposed PDC aching scheme, the need of proposing an efficient partitioning-based caching scheme is felt for Content-Centric Networks. Most of the existing solutions implement hard partitioning-based schemes such as k-split or k-medoid partitioning, which require prior network knowledge to define the number of partitions. It triggers the demand for an efficient network partitioning scheme that can partition the network dynamically and determine its performance using widespread simulations. A mechanism is required that will resolve the content placement issues by jointly considering a few of the network and content parameters. These parameters may be node degree, content popularity, cache capacity, number of cache replacements, the cost to fetch the same content from the server, etc. The proposed research work needs to formulate an efficient caching scheme for CCN with a network partitioning scheme and it should reduce network traffic, hop-count, and content retrieval delay in the network.

4.2 Proposed caching scheme: DPPCOP

The proposed DPPCOP caching strategy partitions the CCN dynamically and uses the Elbow method [105,106] to find the appropriate number of partitions. The network partitioning is performed to control excessive content caching operations and the collaboration among routers increases content diversity in the network. The scheme considers network partitioning, content popularity, and distance-based parameters during caching decisions. Using these parameters, the scheme places popular contents near the requesters with reduced content redundancy and a lesser number of caching operations. For content popularity determination, the scheme uses the dynamically sized Popularity Window as discussed in section 3.3 [103]. By intelligently computing the Popularity Window size based on the number of distinct contents in the network, the scheme improves QoS for the requesters with negligible computational overhead [99]. To evict the older contents from the cache once it becomes full, the DPPCOP scheme uses a widely acknowledged LRU cache replacement algorithm.

4.3 Proposed Network Partitioning Scheme

The proposed partitioning strategy begins with designating each network router as a partition head as shown in the below-mentioned algorithm $Network\_Partitioning(|R|, P_x, P_y)$. Then, the partitioning scheme dynamically merges two non-overlapping partitions that are closest to each other. The scheme uses the bandwidth and hop count characteristics to determine the closeness between the partitions. The distance among the partitions is computed using the "Update_Distance(P_x, P_y)" algorithm and it is updated each time a new partition is formed. For merging of partitions, the scheme uses the "Merge_Partition($|R| - (i-1), P_x, P_y$)" algorithm which is discussed subsequently. To generate an optimal number of partitions in the network, the scheme uses the Elbow method instead of forming the arbitrary number of partitions [107].

Network_Partitioning $(|R|, P_x, P_y)$

Initialize each $\{R_x; 1 \leq x \leq |R|\}$ as P_x with $PH(P_x) = R_x$.

1. Update_Distance(P_x, P_y)

1. for $x = 1$ to $|R|$
2. for $y = 1$ to x
   (a) Compute, $Dist(P_x, P_y)$ using Equation. 4.1.
   (b) $M(P_x, P_y) = Dist(P_x, P_y)$.

2. Merge_Partition $(|R| - (i-1), P_x, P_y)$

1. Determine $WSS_{|R|}$ using Equation 4.2 for initial $|R|$-partitions.
2. Initialize *i=0*.
3. Iterate the following steps:
   (a) *i=i+1*.

(b) Select P_x and P_y, such that $Dist(P_x, P_y)$ is minimum in $M(P_x, P_y)$.

(c) Merge P_x and P_y to create $P_{x,y}$ which would form $(|R| - i)$ partitions in the network.

(d) Determine $WSS_{|R|-i}$ using Equation. 4.2.

(e) If $\left[\left(WSS_{|R|-i} - WSS_{|R|-(i-1)}\right) > \beta\right]$, then

   i. Discard merging of partitions performed in $step - 3(c)$.

   ii. Stop iteration of $step - 3$.

(f) Else,

   i. $PH(P_{x,y}) = R_i$, where $R_i \in P_{x,y}$ and $Dist(R_i, Serv)$ is minimum.

   ii. Execute $Update\_Distance(P_{x,y}, P_z)$ procedure, where $P_{x,y}$ and P_z are disjoint partitions.

In the proposed "Network_Partitioning $(|R|, P_x, P_y)$" algorithm, initially each router (R_x) has been designated as a partition P_x and its own partition head $(PH(P_x))$. During first iteration, the proposed strategy executes "$Update\_Distance(P_x, P_y)$" algorithm to calculate the distance among routers in the network. The $step - 1$ *and* $step - 2$ of "$Update\_Distance(P_x, P_y)$" algorithm takes the average bandwidth among disjoint partitions, bandwidth between P_x and P_y and hop count value for distance determination. The distance values are placed in the matrix $M(P_x, P_y)$. The matrix $M(P_x, P_y)$ is computed every time after the partitions are merged. This is used to select two closest partitions for merge operation until the optimal number of partitions are created. The value of $Dist(P_x, P_y)$ parameters is computed using Equation 4.1 as shown below.

$$Dist(P_x, P_y) = \frac{B}{b\_width(P_x, P_y)} \times H(P_x, P_y) \qquad (4.1)$$

The distance between two partitions has been computed as the ratio of the product of average network bandwidth with the shortest path (hop-count) between two farthest network routers in the partitions and the minimum bandwidth between those routers. The tightly coupled partitions are formed by applying these heuristics. Then, the scheme iteratively selects two closest partitions from $M(P_x, P_y)$ and perform their merging using $Merge\_Partition(P_x, P_y)$ algorithm. As illustrated in $step-1$ of the algorithm, the value of WSS_n (Within Cluster Sum of Square) is computed using Equation 4.2.

$$WSS_n = \sum_{k=1}^{n} \sum_{j=1}^{p} Dist(R_j, PH(R_j))^2 \tag{4.2}$$

Here, n denotes the number of partitions in the network. The variable p defines the number of network routers in the k^{th} partition. Initially, when each router itself is a partition head that is *p=1*, the value of $WSS_{|R|}$ would be *0*. In the next iteration, when the two closest partitions merge, the number of remaining partitions are $(|R|-1)$ and the value of $WSS_{(|R|-1)}$ is computed as 1.

In $step-3(a)-3(c)$ of $Merge\_Partition(P_x, P_y)$ algorithm, two closest partitions P_x and P_y are selected from $M(P_x, P_y)$ and are merged to create $P_{x,y}$. The iterations of these steps decrease the number of partitions in the network. In $step-3(d)$, the value of the expression $WSS_{(|R|-i)}$ is computed. The $step-3(e)$ computes the change in the value of the WSS (Within-cluster Sum of Squares) [108] each time when the number of partitions are reduced. A curve has been plotted for these values *w.r.t* to the number of partitions. Equation 4.2 shows the expression for WSS computation [109]. Initially, the value of WSS does not increase rapidly with the merging of partitions but it demonstrates a rapid increase after the creation of a specific number of partitions. This forms an "Elbow" in the graph and this

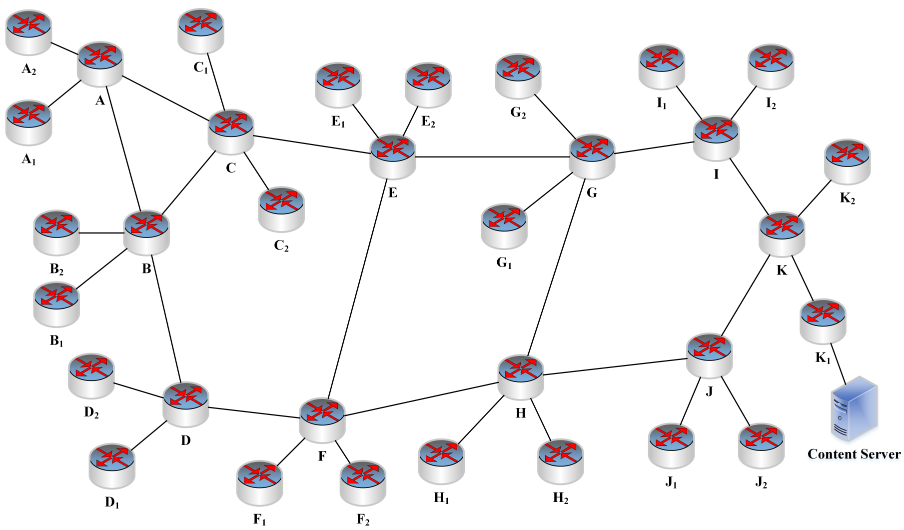


Figure 4.1: Partial illustration of Abilene network topology

bend is used as an indicator of the "optimal" number of partitions. For this, the scheme determines the value of $WSS_{(|R|-i)}$ each time after the merge operation and compute the gain achieved from the value of $WSS_{(|R|-(i-1))}$ $(step-3(e))$. The gain greater than β indicate that the "Elbow" is formed and therefore, the newly created partition need to be discarded and merging of partitions stops with $(|R|-(i-1))$ partitions. Otherwise, if the gain in WSS is less than β after forming new partitions, then the intra-partition router which is nearest to the cloud server is designated as the new partition head $(step-3(f-i))$. Before next iteration of merging of partitions, the distance between newly formed partition and the other partitions is determined using "Update_Distance(P_x, P_y)" algorithm and a new matrix $M(P_x, P_y)$ is formed $(step-3(f-ii))$.

4.4 An illustration of the proposed network partitioning scheme

An illustration of the Abilene network topology [110,111] has been shown in Fig. 4.1 to explain the working of the proposed network partition scheme.

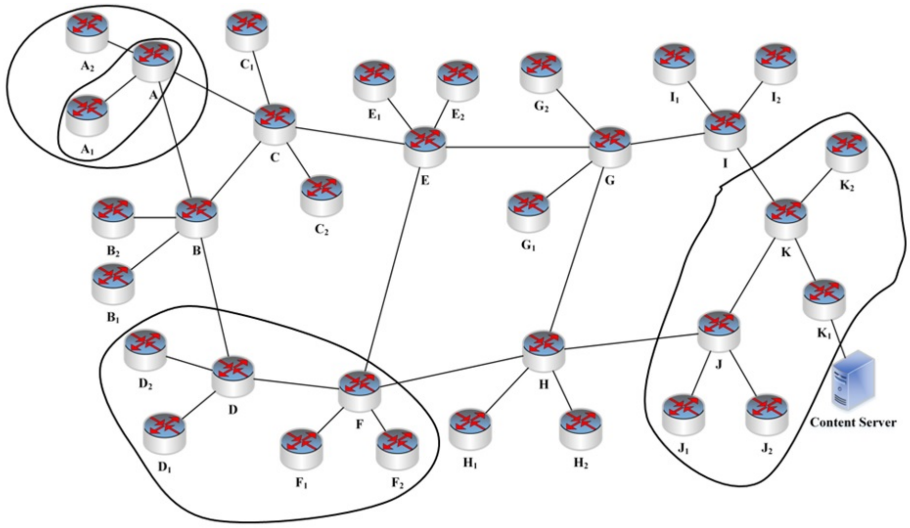


Figure 4.2: Illustration of partitioning in the partial Abilene network topology

As per $step-1$ of the proposed partitioning scheme, initially, the network routers $(A, A_1, A_2, B, ..., K_2)$ are designated as individual partitions in the network. To simplify the discussion, suppose the bandwidth of all connections in the network is identical. According to $step-2$ and $step-3$, for identical bandwidth, the matrix $M(P_x, P_y)$ would be computed as the hop-count among the network routers. Then, the merging of routers in the partitions is performed based on the $Merge\_Partition(P_x, P_y)$ procedure of the partitioning scheme that merges two nearest partitions iteratively.

Fig. 4.2 shows an example of the working of the proposed caching scheme. Initially, router A and A_1 are merged to form a new partition. Then, according to the proposed network partitioning scheme, the routers $\{D, D_1, D_2, F, F_1, F_2\}$ are merged to form a partition later during the execution of the algorithm.

Suppose, the increase in $WSS_{(|R|-i)}$ is assumed to be rapid if its value is improved by $(\beta > 30)$ after decreasing the number of partitions. With merging of partitions iteratively, when the number of partitions become reduced to 4, the WSS_4 is computed as 105. Subsequently, when the number of partitions further reduced to 3, the value of WSS_3 increased to

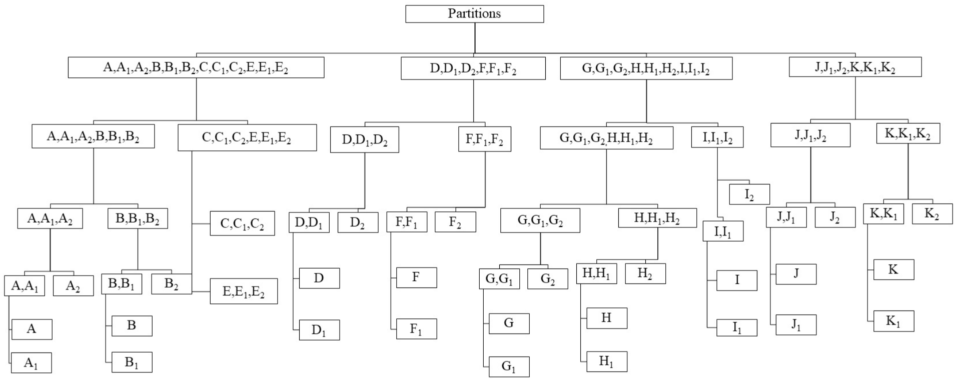


Figure 4.3: Partitions for the network mentioned in Fig. 4.1

148. Therefore, the gain in the value of WSS is greater than 30 ($WSS_3 - WSS_4; \beta > 30$). Hence, merging of the partitions would be ignored and the optimal number of partitions are achieved as 4 with *partition heads* $- E, F, I, K_1$. For the network mentioned in Fig. 4.1, the categorization of routers in each partition are shown in Fig. 4.3.

4.5 Updated structure of Interest and Content messages

In order to make caching decisions in the partitions, the proposed caching strategy considers the partition's information, bandwidth, hop count, and content popularity parameters. Therefore, the structure of the Interest and Content messages needs to be changed to store the information of these parameters. The updated structures of the messages are mentioned below:

4.5.1 Modified structure of Interest message

The below-mentioned fields are embedded in the Interest message to assist during caching of the matching content. Here, the $Dist(I_i)$ field stores the total distance traversed by the Interest message.

$C\_name(I_i)$	$Dist(I_i)$	...

Structure of Interest message

$C\_name(D_i)$	$H(I_i)$	$Dist(I_i)$	$H(D_i)$	$Dist(D_i)$
$S.PH(D_i)$	$O.PH(D_i)$	$\phi(D_i)$	$Payload$	...

Structure of Content message

4.5.2 Modified structure of Content message

The structure of the Content message is modified and has the following additional fields; $H(I_i)$, $Dist(I_i)$, $H(D_i)$, $Dist(D_i)$, $S.PH(D_i)$, $O.PH(D_i)$ and $\phi(D_i)$ along with requested payload. The value of $Dist(I_i)$ is replicated from the Interest message (I_i) to D_i. The $S.PH(D_i)$ field contains the partition head name of the content provider that has prepared D_i. During forwarding of D_i, the $O.PH(D_i)$ field stores the name of the partition head of intermediate in-network routers. The $\phi(D_i)$ field has a boolean value to enable and disable the content caching in the partitions to minimize the content caching operations.

4.5.3 Interest Message Handling Procedure

Initially, the requester U_k creates a message I_i to request the content and transmit it in the network. To process I_i, the in-network router (R_j) executes $procedure - 1$ (Interest message handling procedure (I_i, R_j, U_k)) as shown below. In $step - 1$ and $step - 2$ of the procedure, the CS of R_j is searched for the matching content and if found, then *Content message handling procedure* is executed as discussed in section 4.5.4. If no matching content exists, then the information of the incoming Interest message is stored in the Popularity Window as described in $step - 3(a) - (c)$. The distance traversed by I_i is computed in $step - 3(d)$ using the bandwidth and hop count parameters. This value is stored in the $Dist(I_i)$ field of the Interest message (I_i) which is used during content caching decisions.

$Step-4$ to $step-6$ depicts the traditional Interest message processing steps which involve maintaining entries in the FIB and PIT before Interest message forwarding to the upstream routers. If the router does not find the suitable upstream router to forward I_i, then it would be removed from the network.

Procedure-1: Interest message handling procedure (I_i, R_j, U_k)

Perform following operations on arrival of Interest message (I_i) to R_j from U_k.

1. On arrival of an Interest message (I_i), verify the following equation:

$$H_{R_j}^{I_i} = \begin{cases} True, & If C\_name(I_i)\ exists\ in\ CS\ of\ R_j \\ False, & Otherwise \end{cases}$$

2. *If* $H_{R_j}^{I_i} = True$, then go to *Content message handling procedure* mentioned in section 4.5.4.

3. *Else*

   (a) *If* $W_{R_i}^s = 0$ then,

   $Initialize\ s = 1\ for\ router\ R_j$

   $Move\ S_{R_i}^s \leftarrow C\_name(I_i)$

   (b) *If* $Max(W_{R_i}^s) > (W_{R_i}^s)$ *then,*

   set s=s+1 for router R_j

   Move $S_{R_i}^s \leftarrow C\_name(I_i)$

   (c) *If* $Max(W_{R_i}^s) = (W_{R_i}^s)$ *then*

   do l=2 to $Max(W_{R_i}^s)$

   Move $S_{R_i}^{l-1} \leftarrow S_{R_i}^l$

   $Move\ S_{R_i}^l \leftarrow C\_name(I_i)$

   (d) Update $Dist(I_i)$ field of I_i using following computation

$H(I_i)$=$H(I_i)$+1.

$Dist(I_i)$= $\dfrac{B}{b\_width(R_j, U_k)} \times H(I_i)$.

4. If I_i exists in $PIT(R_j)$, then aggregate I_i in PIT(R_j) and dispose I_i.
5. Else, if $FIB(R_j)$ has upstream router/server information for I_i forwarding, then forward I_i to upstream router/server accordingly.
   (a) Make record of I_i in $PIT(R_j)$.
6. Else, dispose I_i from the network.

4.5.4 Content Message Handling Procedure

When a matching content ($C\_name(I_i)$) is found in the CS of router (R_j) or I_i needs to be served by the server, the router/server becomes content provider. The provider prepares a corresponding Content message (D_i) as discussed in the *"Content message handling procedure"*. For this, the content provider (R_j) performs $step - 1$ of the procedure as defined below.

Procedure-2: Content message handling procedure
(D_i, R_j, R_m, U_k)

1. If requested content I_i exists in the $CS(R_j)$ then
   (a) Create Content message (D_i) corresponding to I_i.
   (b) Copy the value of $Dist(I_i)$ field from I_i to D_i.
   (c) Set the value of field, $Dist(D_i)$ =0.
   (d) Set the name of $PH(R_j)$ in the $S.PH(D_i)$ *and* $O.PH(D_i)$ fields of Content message (D_i).
   (e) Reset the field, $\phi(D_i) = 0$.
   (f) Forward D_i in the reverse direction towards U_k.

2. Else, when Content message is received at the intermediate router R_m from the content provider (R_j)/server then perform $step - 3$ to $step - 7$.

3. Compute and overwrite the value in the distance field $Dist(D_i)$ as the distance between R_j and R_m.

   $H(D_i) = H(D_i) + 1.$

   $Dist(D_i) = \dfrac{B}{b\_width(R_j, R_m)} \times H(D_i).$

4. If, R_m is a partition head then,

   (a) Overwrite its name (R_m) into $O.PH(D_i)$ field of the D_i.

   (b) If $S.PH(D_i)! = O.PH(D_i)$ then

   Set the field, $\phi(D_i) = 1$.

5. $Placement\_Score = W_{R_m}(C\_name(D_i)) \times \dfrac{Dist(D_i)}{Dist(I_i)}$

6. If $\phi(D_i)$=1 & $O.PH(D_i) = PH(R_m)$ & $S.PH(D_i)! = PH(R_m)$ & $Placement\_Score \geq T_R$, then

   (a) Cache content in the $CS(R_m)$ using the LRU replacement mechanism.

   (b) Reset the $\phi(D_i)$ field to 0.

7. Forward D_i towards requester U_k.

In $step - 1$, the router R_j move the value of $Dist(I_i)$ field from I_i to the identical field in D_i and set the value of $Dist(D_i)$ to 0. Then, the provider (R_j) insert its partition-head name $PH(R_j)$ in the $S.PH(D_i)$ and $O.PH(D_i)$ fields of D_i . The field $\phi(D_i)$ is initialized with 0. In the proposed scheme, $\phi(D_i) = 0$ and $\phi(D_i) = 1$ define that the content caching is disabled and vice-versa. The generated Content message D_i is then forwarded towards the requester (U_k).

The on-path router that receives the Content message D_i, performs $step - 3$ to $step - 7$ for content caching and forwarding operations. In

$step-3$, the router computes the $Dist(D_i)$. The proposed caching strategy caches the contents in the intermediate partitions only when the content is passed through their partition heads. When the content is found within the same partition that generated the Interest message, then the content is routed to the requester without caching operations. These characteristics minimize the number of cache replacement operations and ensure high content diversity within the partition. $Step-4$ is performed to enable content caching in the intermediate partition when content is passed through the partition head. Then, the value of $Placement\_Score$ is computed based on the content popularity and normalized distance parameters ($step-5$). The content is cached in the intermediate router, if D_i is forwarded from another partition, its caching is enabled ($\phi(D_i) = 1$) and the $Placement\_Score$ is greater than or equal to the threshold value (T_R). If the content is cached, then the value of $\phi(D_i)$ is reset to 0 to avoid excessive caching of D_i within remaining intra-partition routers ($step - 6$). Finally, irrespective of the caching decision, the content is forwarded towards the requester (U_k) as mentioned in $step - 7$.

4.6 Simulation parameters and setup

The simulation environment is used to produce the real-time architecture of the CCN. There are various simulation tools for CCN such as ndnSIM [112, 113], ccnSIM [114], Icarus [90], and OMNeT++ [82, 115] etc. Among these tools, the ndnSIM is a widely accepted and open-source simulation tool, and therefore, it is used to analyze the performance of the caching schemes. The ndnSIM allows programmable changes using the C++ programming language. Using the ndnSIM tool, the programmers can customize the evaluation parameters and network configurations. Hence, the tool is deployed on the Ubuntu 16.04 operating system to evaluate the performance of proposed PDC and DPPCOP strategies and existing caching schemes that are considered in the thesis.

To evaluate the network performance, the US (United States)-based Abilene network topology has been implemented in the ndnSIM simulation environment with 167 nodes. The Abilene network topology is used for communication among Universities and few corporate and affiliated organizations in all of the US states along with the District of Columbia and Puerto Rico. The 167 network nodes are categorized into 1 content server, 33 in-network routers, and 133 requesters. Initially, the content server holds the content catalog to serve the Interest messages. The in-network routers have the content caching capability and are also used to forward the Interest message and Content messages towards providers and requesters respectively. The requesters generate the Interest messages in the network and they are the ultimate recipient of the Content messages.

4.6.1 Simulation parameters for PDC caching scheme

For realistic simulation results in the proposed PDC caching scheme, the value of content catalog size ($|D|$) is set to 5000 contents of uniform payload size. The cache size ($C_{size}(R_i)$) of each in-network router is set to $1-2\%$ of the catalog size i.e. 50/100 contents in different simulation executions and therefore, ($C_{size}(R_i)$) $<<$ $|D|$. The Content messages carry 1 KB of *payload* which are generated in response to the Interest messages. The value of the popularity skewness parameter α is 0.7 to analyze a wide range of content popularity distribution patterns. The request generation rate is set to $50/second$ for each requester and therefore, approximately ($133 \times 50 =$) 6650 Interest messages are generated every second in the network. The LRU strategy is implemented as the default cache replacement scheme for the proposed and existing caching schemes.

Table 4.1: Simulation parameters values

Parameter	Value
$\|R\|$	33
Total number of requesters	133
Content server	1
$\|D\|$	5000
α	0.7
$C_{size}(R_i)$	50/100
λ	50 per second
δ	0.6
Connection delay	10ms
Connection bandwidth	Up to 10Mbps
θ	0.02
Payload size	1 KB
Simulation duration	1050 STU (Simulation Time Unit)

The value of "θ" plays a critical role in computing the content demand in the network routers. It has been observed that increasing the value of "θ" beyond a certain point excessively increases the computational latency and does not improve the network performance significantly. Therefore, the value of "θ" has been taken as 0.02 to compute the size of the Popularity Window. Using this value, the Popularity Window can store up to ($\lfloor 0.02 \times 5000 \rfloor$, *as per Equation* 3.4) 100 content names in the proposed caching schemes when network have 5000 distinct type of contents. The link delay among the network nodes is kept at 10 ms with network bandwidth up to 10 Mbps respectively. The simulation parameters and their corresponding values used in the PDC caching strategy are summarized in Table. 4.1.

4.6.2 Simulation parameters for DPPCOP caching scheme

In the proposed DPPCOP caching scheme, the in-network routers of the Abilene network are divided into several partitions as shown in Fig. 4.3. When the value of β is set to 30 during network partitioning, the network routers are eventually divided into 4 partitions using the Elbow method. For performance evaluation of the DPPCOP scheme, the values of the

Table 4.2: Simulation parameters values

Parameter	Value
$\|R\|$	33
Total number of requesters	133
Content server	1
$\|D\|$	5000
α	0.8 and 1.1
$C_{size}(R_i)$	50
λ	50 per second
β	30
Connection delay	10ms
Connection bandwidth	10Mbps
θ	0.1
Payload size	1 KB
Simulation duration	150 seconds

simulation parameters are depicted in Table 4.2.

After network partitioning, the performance of the proposed caching scheme is examined for a wide variety of threshold values (T_R) in terms of the cache hit ratio characteristic. The cache hit ratio for an in-network router is computed as the ratio of the total number of cache hits in the cache and the total number of Interest messages received by the router. Fig. 4.4 illustrates the cache hit ratio achieved by the proposed DPPCOP caching scheme in the entire network for different threshold values ($T_R = 1\ to\ 7$). The optimal cache hit ratio is obtained when $T_R = 2$ with $\alpha = 0.8$ and therefore, these simulation values are considered during execution of the proposed caching scheme. Although the values of β and T_R are relatively arbitrary and may differ for other network topologies, it offers a sufficiently acceptable foundation point to evaluate the performance of large-scale Content-Centric Networks.

4.7 Evaluation parameters & Criteria

The following evaluation parameters are used to examine the performance of the proposed PDC and DPPCOP caching schemes with peer caching strategies.

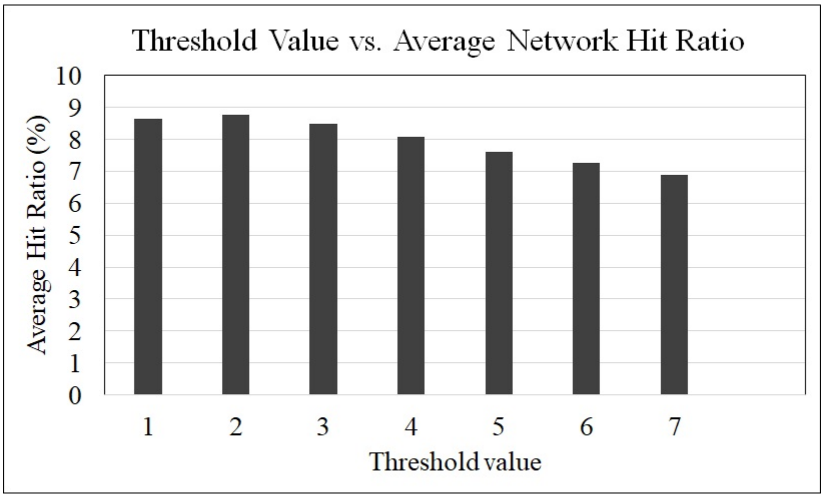


Figure 4.4: Computation of (T_R) in the DPPCOP caching scheme

4.7.1 Cache hit ratio

The cache hit ratio is one of the key indicators to explore the caching performance in the network. A cache hit occurs when the content is found in the CS (Content Store) of the router. Otherwise, if the content does not found in the CS, then it is called a cache miss. The division of cache hit to the total number of Interest messages received by the in-network routers is represented using the cache hit ratio. Equation 4.3 [116] shows the heuristics to compute the value of this parameter.

$$Hit\_Ratio = \frac{\sum_{i=1}^{|R|} Hit(R_i)}{\sum_{i=1}^{|R|} Hit(R_i) + \sum_{i=1}^{|R|} Miss(R_i)} \tag{4.3}$$

As defined in Equation 4.3, the cache hit ratio improves when the requested contents are found near the requesters and the number of cache miss operations reduces in the network. Hence, the performance of the network increases with an increase in the accuracy of content placement decisions.

4.7.2 Average network hop count

This evaluation criterion determines the total number of hops pass-over by the Interest messages and the corresponding Content messages to reach the content provider and the requester respectively. A decrease in the value of the average network hop count represents improved QoS for the requesters as the content is accessed from the nearby network routers. Equation 4.4 shows the value of average network hop count observed by an Interest message (I_j) and its matching Content message (D_j) during content retrieval.

$$Hop\_Count = H(I_j) + H(D_j) \tag{4.4}$$

4.7.3 Average network delay

The network delay for a requested content is computed as the time duration between Interest message generation and the receiving of the matching Content message. This metric also adds the Interest message retransmission delay if the content is not retrieved within a defined period. The average network delay is a significantly different performance criterion from the average network hop count as it also includes the computational delay experienced during content caching decisions. During simulations, the value of average network delay is computed in terms of micro-seconds (μs).

4.7.4 Average network traffic

The average network traffic is computed as the aggregation of network traffic (in KB) detected on the links (E) in per unit time (second). This parameter denotes the exploitation of available network bandwidth and the computational resources of in-network routers. When the cache hit ratio increases, the contents are retrieved with lesser delays and it also decreases the network traffic. Thus, the network traffic tends to decrease with improvement in the accuracy of content placement decisions.

4.8 Summary

In this chapter, the PDC and DPPCOP caching schemes are proposed for the content placement decisions. The next chapter will present the performance evaluation and analysis of the proposed caching strategy. The next chapter will also compare the performance of suggested PDC and DPPCOP schemes with traditional routing protocols.

Chapter 5

Result Discussions and Analysis

This chapter discusses the performance results achieved by the proposed PDC and DPPCOP caching schemes and compares them with existing caching strategies. The simulation results are compared based on the hit ratio, average network hop count, average delay, and network traffic metrics. These results are obtained for different caching capacities and a wide range of content popularity patterns in a realistic network configuration.

5.1 Result Discussions and Analysis of the Proposed PDC Caching Scheme

In this section, the performance of the proposed PDC caching scheme is presented. The results show significant performance gain from the peer caching strategies on different QoS parameters as follows:

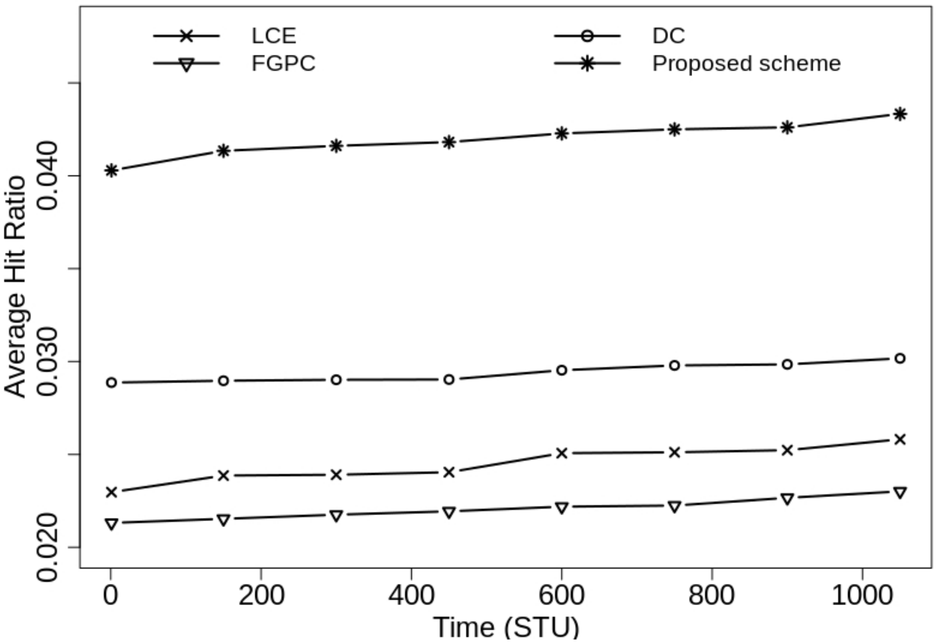


Figure 5.1: In-network cache hit-ratio of caching schemes with C_{size}=50, $|D| = 5000$, $\alpha = 0.7$ and $\lambda = 50/second$

5.1.1 Effect on cache hit ratio (PDC caching scheme)

The cache hit ratio defines the utilization of cache space to serve the requested contents in the network. It is computed using Equation 4.3. The increase in the value of the cache hit ratio indicates improved cache space utilization and QoS for the requesters. As shown in Fig. 5.1, when C_{size}=50, the proposed PDC caching scheme shows improved cache hit ratio during simulations as compared to the traditional caching scheme (LCE), DC-based, and FGPC caching schemes.

As the proposed PDC caching scheme considers content popularity and distance parameters during content placement decisions, the scheme is able to cache popular contents near the requesters. Therefore, a similar performance gain is achieved when the caching capacity is increased to 100 (2% of the content catalog size), and it is illustrated in Fig. 5.2. As shown in Fig. 5.2, the proposed PDC caching scheme outperforms the existing caching strategies by showing up to 1.7%, 1.1%, and 2.0% gain in the cache hit ratio

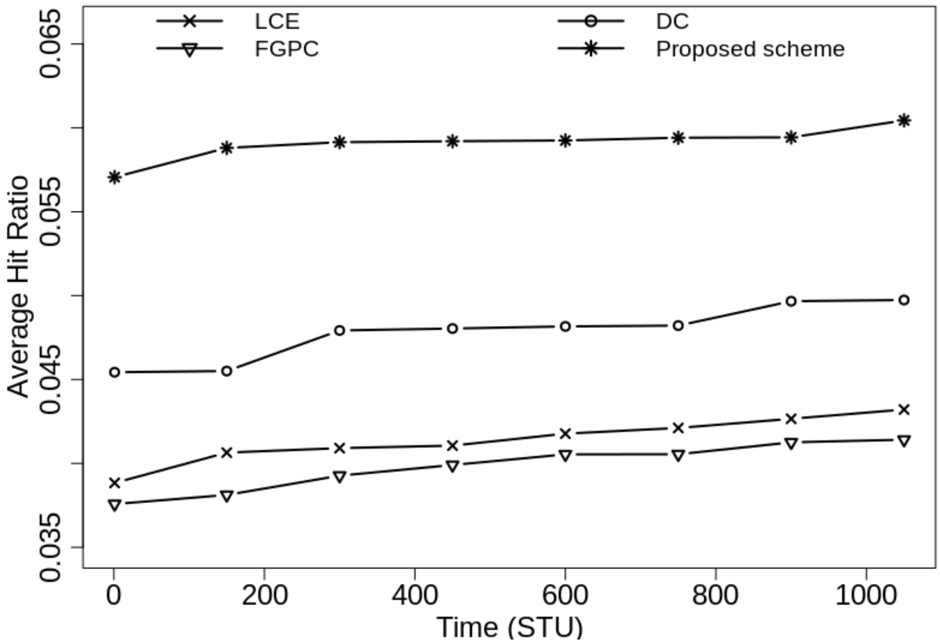


Figure 5.2: In-network cache hit-ratio of caching schemes with C_{size}=100, $|D| = 5000$, $\alpha = 0.7$ and $\lambda = 50/second$

Table 5.1: Gain of proposed caching scheme on existing schemes based on cache hit ratio

Gain in cache hit ratio			
Network configuration (C_{size},$\lvert D\rvert$, α, λ)	**PDC Vs. LCE**	**PDC Vs. DC**	**PDC Vs. FGPC**
(50,5000,0.7,50)	1.75%	1.3%	2.0%
(100,5000,0.7,50)	1.7%	1.1%	2.0%

from the LCE, DC-based, and the FGPC caching strategies respectively.

The hit ratio gain has been computed as the difference between the cache hit ratio of the proposed and existing caching scheme. It is computed using Equation 4.3. Table 5.1 summarizes the hit ratio gain achieved by the PDC scheme over existing schemes for different caching capacities.

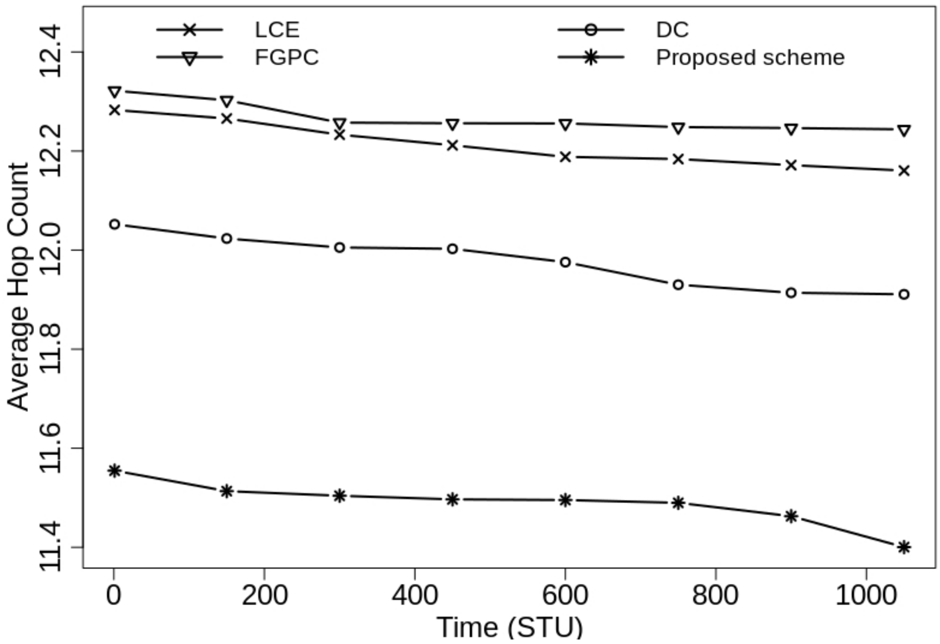


Figure 5.3: Average network hop-count of caching schemes with C_{size}=50, $|D| = 5000$, $\alpha = 0.7$ and $\lambda = 50/second$

5.1.2 Effect on average network hop count (PDC caching scheme)

The average network hop count parameter defines the number of hops traversed by the messages to access the contents in the network. The decrease in hop count value indicates improved network performance as the content is retrieved with reduced delay. Fig. 5.3 shows the average network hop count value observed under different caching schemes when $C_{size} = 50$. In this scenario, the proposed PDC scheme shows up to a 6.9% decrease in average network hop count from the existing caching schemes.

Analogous performance results are discovered when the caching capacity is set to 100 contents for each in-network router. In this scenario, the proposed scheme outperforms the existing caching schemes by achieving up to 5.4%, 3.3%, and 6.0% decrease in the hop count value from LCE, DC-based, and FGPC caching solutions.

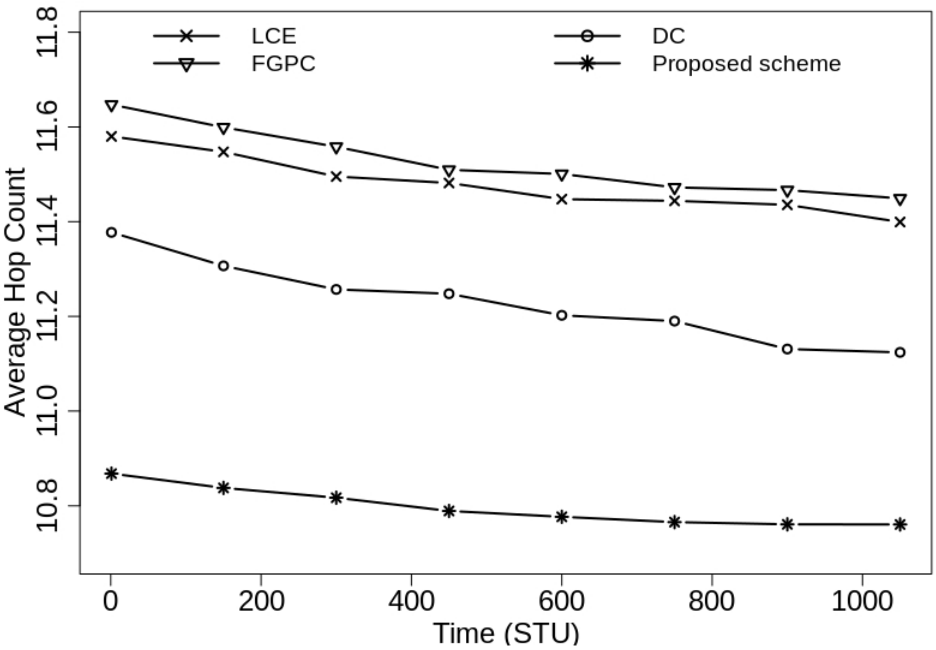


Figure 5.4: Average network hop-count of caching schemes with C_{size}=100, $|D| = 5000$, $\alpha = 0.7$ and $\lambda = 50/second$

$$\%HC\_Reduction = \frac{(Hop\_Count(ES) - Hop\_Count(PS)) \times 100}{Hop\_Count(ES)} \tag{5.1}$$

The percentage of hop count reduction is computed using Equation 5.1. Here, $\%HC\_Reduction$, $Hop\_Count(ES)$ and $Hop\_Count(PS)$ represent the percentage of reduction in hop count, the number of hops observed under the existing caching scheme, and hop count experienced in the proposed caching scheme respectively. Table 5.2 summarizes the reduction in the average network hop count for PDC scheme from existing schemes under different caching capacities.

Table 5.2: Average hop count reduction in the proposed PDC caching scheme from existing schemes

Reduction in average network hop count			
Network configuration (C_{size},$\|D\|$, α, λ)	**PDC Vs. LCE**	**PDC Vs. DC**	**PDC Vs. FGPC**
(50,5000,0.7,50)	6.3%	4.2%	6.9%
(100,5000,0.7,50)	5.4%	3.3%	6.0%

5.1.3 Effect on average network delay (PDC caching scheme)

The average network delay is determined in micro-seconds and this metric shows the average latency in content retrieval from the Interest message generation. Therefore, lesser network delay suggests efficient utilization of the available caching resources. Fig. 5.5 shows the average network delay experienced under different caching schemes when $C_{size} = 50$. As shown in the Figure, the average network delay is $\approx 112000ms(micro-seconds)$ for the proposed PDC caching scheme and it outperforms the existing LCE, DC-based, and the FGPC caching strategies.

When caching capacity of the in-network routers is increased and set to 100 with keeping other parameters remain unchanged, the average network delay reduces for all caching strategies. As shown in Fig. 5.6, in this scenario also the PDC scheme significantly reduces the average network delay as compared to peer schemes. For these simulation executions, the percentage reduction in average network delay is summarized in Table 5.3.

5.1.4 Effect on average network traffic (PDC caching scheme)

The average network traffic determines the utilization of available network bandwidth and indicates the load on communication links. The decrease

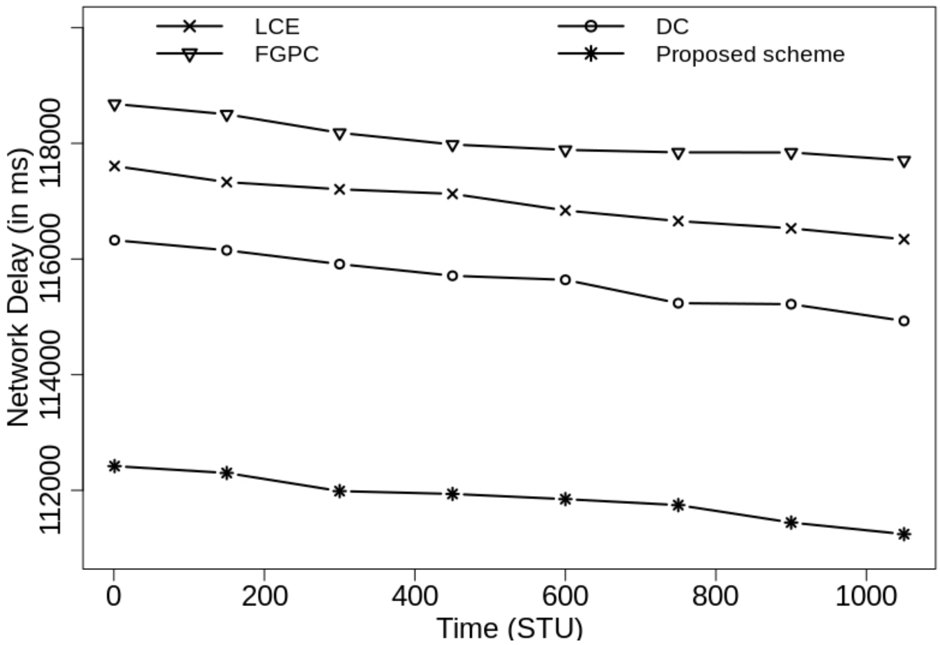


Figure 5.5: Average network delay (in micro-seconds) with C_{size}=50, $|D| = 5000$, $\alpha = 0.7$ and $\lambda = 50/second$

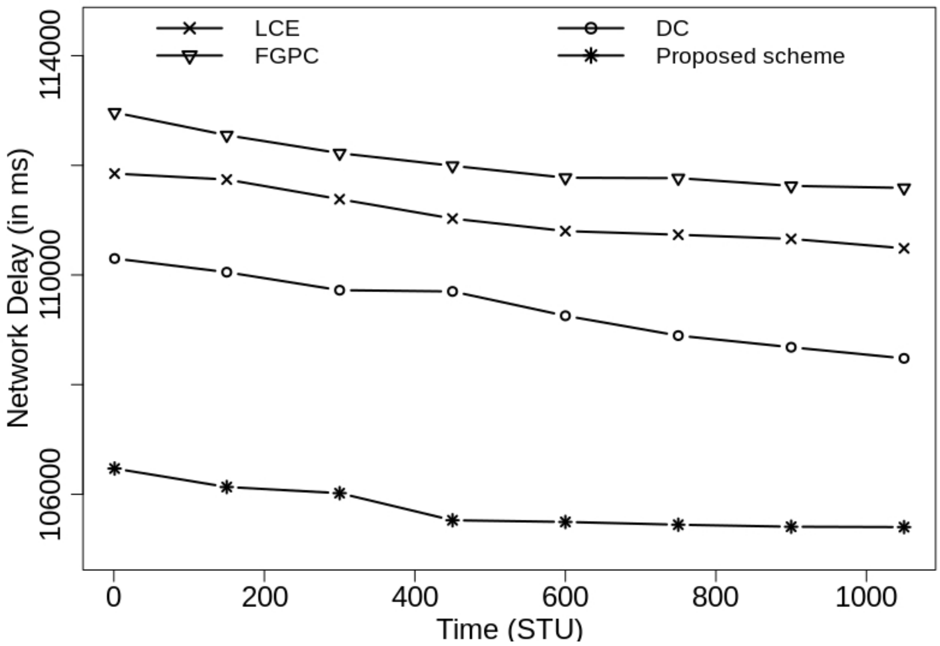


Figure 5.6: Average network delay (in micro-seconds) with C_{size}=100, $|D| = 5000$, $\alpha = 0.7$ and $\lambda = 50/second$

Table 5.3: Average network delay reduction in the proposed PDC caching scheme from existing schemes

Reduction in average network delay			
Network configuration (C_{size},$\|D\|$, α, λ)	**PDC Vs. LCE**	**PDC Vs. DC**	**PDC Vs. FGPC**
(50,5000,0.7,50)	4.4%	3.2%	5.5%
(100,5000,0.7,50)	4.6%	2.8%	5.6%

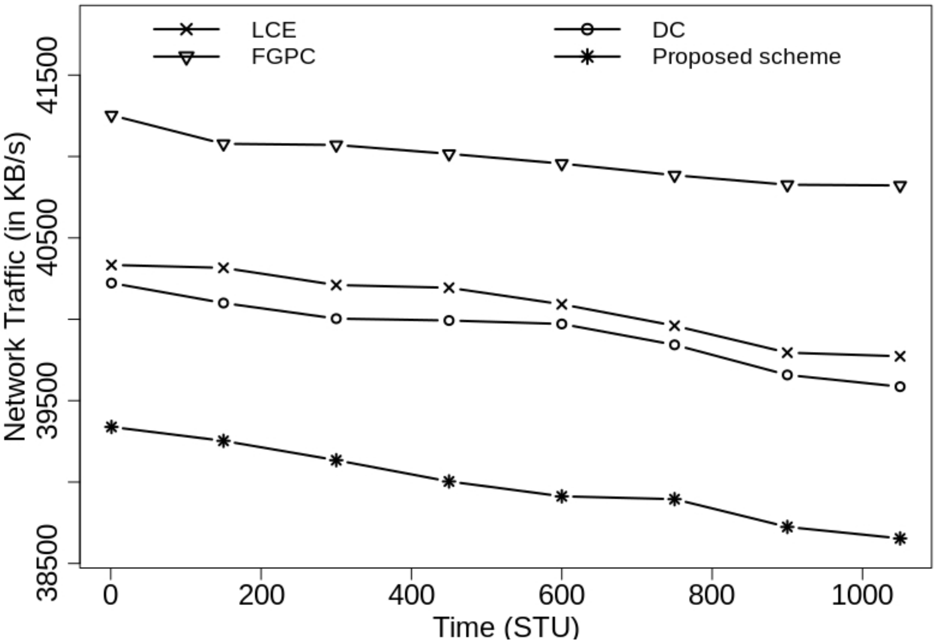


Figure 5.7: Average network traffic (in KB/second) with C_{size}=50, $|D| = 5000$, $\alpha = 0.7$ and $\lambda = 50/second$

in link load leads to lesser congestion in the network and improves the scalability of the network. Therefore, a decrease in average network traffic improves network performance and QoS for the requesters. This metric is measured in KB/s.

Fig. 5.7 and Fig. 5.8 demonstrate the average network traffic for different caching schemes when C_{size} is 50 and 100 respectively. The results demonstrate that the PDC caching scheme decreases the network traffic as compared to the existing caching schemes in both simulation scenarios.

The simulation results also illustrate that the network traffic and cache

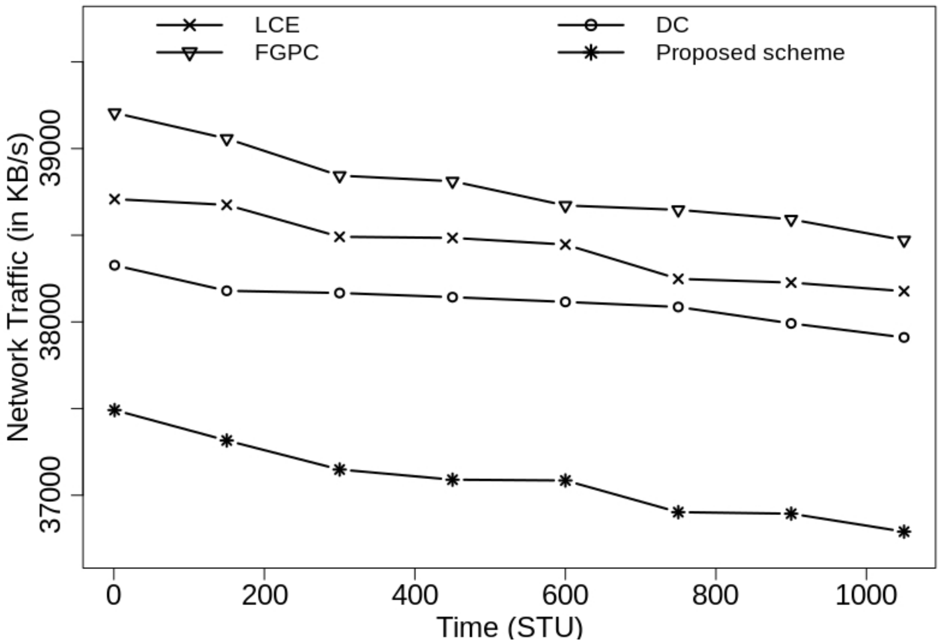


Figure 5.8: Average network traffic (in KB/second) with C_{size}=100, $|D| = 5000$, $\alpha = 0.7$ and $\lambda = 50/second$

Table 5.4: Average network traffic reduction in the proposed PDC caching scheme from existing schemes

Reduction in average network traffic			
Network configuration (C_{size},$\|D\|$, α, λ)	**PDC Vs. LCE**	**PDC Vs. DC**	**PDC Vs. FGPC**
(50,5000,0.7,50)	2.7%	2.4%	4.9%
(100,5000,0.7,50)	2.9%	3.6%	6.3%

hit ratio has an inverse relation among them. When the cache hit ratio decreases, more contents are served by the server which increases network traffic and vice-versa. The percentage of reduction in average network traffic is described in Table. 5.4 for C_{size}=50 and 100.

5.2 Result Discussions and Analysis of the Proposed DPPCOP Caching Scheme

In this section, the performance of the proposed partitioning-based DPP-COP caching scheme is presented and compared with state-of-the-art existing caching strategies. Table 4.2 summarizes the values of simulation parameters used to obtain the results. As discussed in section 4.4, the Abilene network is partitioned before performing the caching operations. After partitioning of the network, the caching decisions are taken based on the partition information, content popularity, and distance parameters as illustrated in section 4.5.3 and 4.5.4. During performance evaluation, the performance of the proposed DPPCOP caching scheme is compared with the traditional caching schemes such as LCE, LCD, and recent strategies like DC-based, FGPC, and the CPNDD schemes.

5.2.1 Effect on cache hit ratio (DPPCOP caching scheme)

Fig. 5.9 and Fig. 5.10 evaluate the cache hit ratio of the DPPCOP caching strategy with existing schemes when $\alpha = 0.8$ and $\alpha = 1.1$ respectively. Initially, all the caching schemes show a low cache hit ratio till 25 seconds as the in-network cache are empty and begin caching of the incoming contents. After 25 seconds, the hit ratio of caching schemes improves as more contents are retrieved from the CS of in-network routers and a lesser number of contents need to be served by the server.

When $\alpha = 0.8$, the DPPCOP caching scheme outperforms the existing caching schemes as shown in Fig. 5.9. The gain in the hit ratio is achieved due to improved cache diversity in the network and caching of popular contents near the requesters. During simulations, the DPPCOP scheme

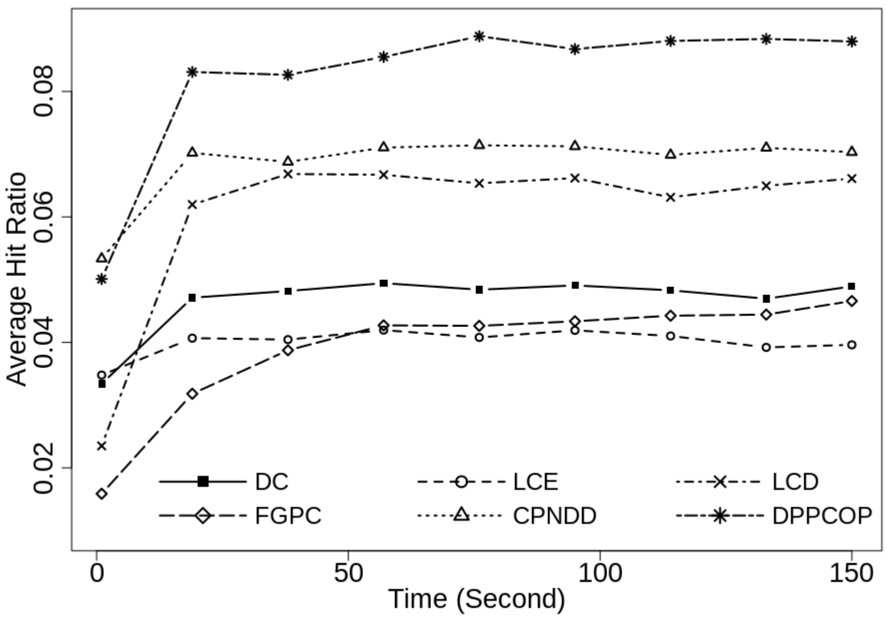


Figure 5.9: Average network hit ratio with C_{size}=50, $|D| = 5000$, $\alpha = 0.8$ and $\lambda = 50/second$

shows up to 4.8%, 2.2%, 3.9%, 4.1%, and 1.8% gain in hit-ratio from the existing LCE, LCD, DC-based, FGPC, and CPNDD caching strategies.

The hit ratio of all the caching schemes improves when the value of α in Zipf distribution is increased to 1.1 while keeping other simulation parameters remain unchanged as shown in Fig. 5.10. This improved cache hit ratio is obtained as a large subset of content requests are concentrated on a smaller subset of the content catalog. Therefore, popular contents are placed within different partitions with increased probability which results in an increased cache hit ratio. As illustrated in Fig. 5.10, the DPPCOP scheme shows an improved cache hit ratio up to 8.6% from the existing caching strategies.

5.2.2 Effect on average network hop count (DPPCOP caching scheme)

When the cache hit ratio improves in the network, it indicates that the contents are accessed from the nearby routers with lesser cache miss prob-

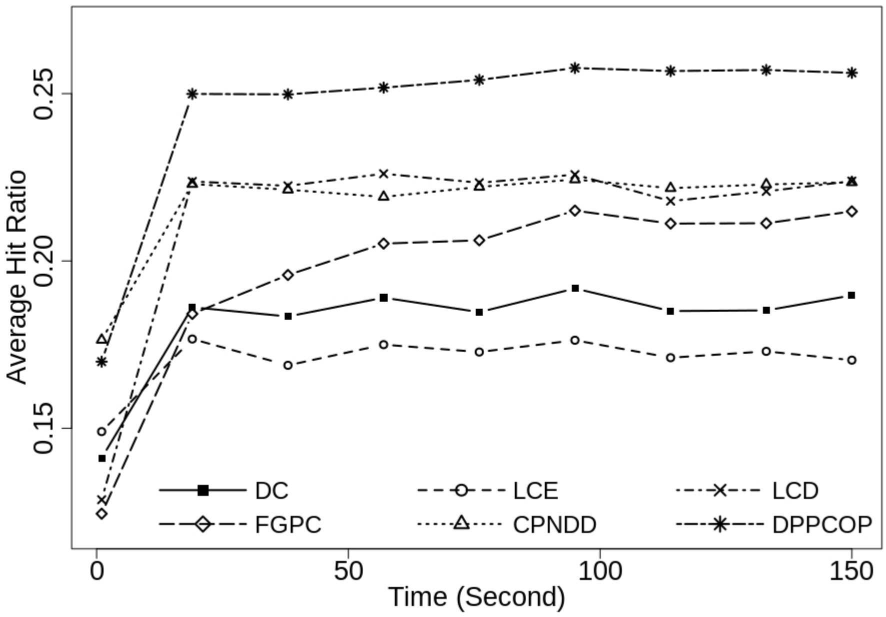


Figure 5.10: Average network hit ratio with C_{size}=50, $|D| = 5000$, $\alpha = 1.1$ and $\lambda = 50/second$

ability. Therefore, the average network hop count decreases with increase in the cache hit ratio. Fig. 5.11 shows the average network hop count for the DPPCOP and the existing caching schemes. As demonstrated, the DPPCOP scheme reduces the average network hop count significantly as compared to peer caching schemes.

A similar reduction in hop count is shown in Fig. 5.12 when the exponent value of Zipf distribution is increased to 1.1. In this scenario, the network hop count for all the caching schemes improves significantly due to the increased number of Interest messages for the smaller number of contents. The simulation results show that the proposed DPPCOP scheme reduces the average network count up to 21.3%, 6.4%, 16.2%, 7.9%, and 8.6% from the LCE, LCD, DC-based, FGPC, and CPNDD strategies respectively.

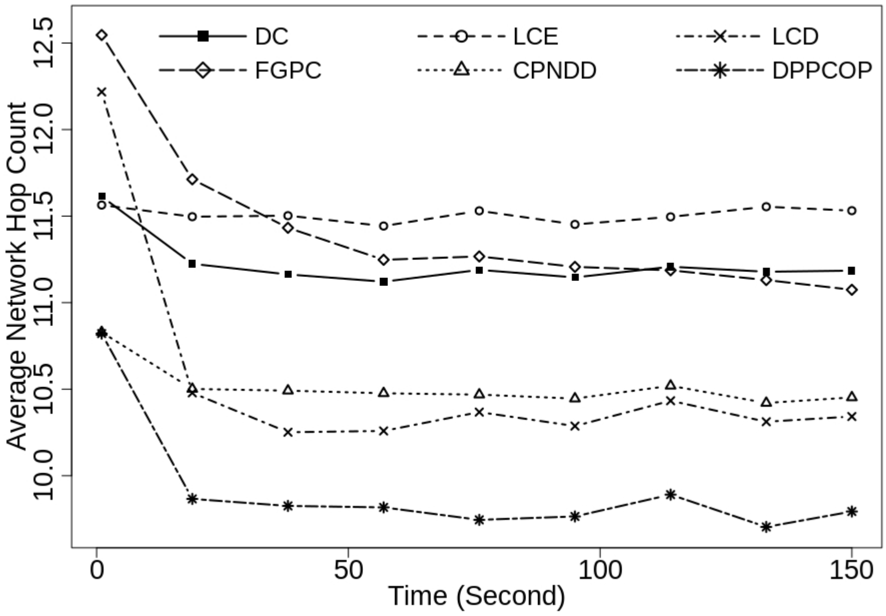


Figure 5.11: Average network hop-count with C_{size}=50, $|D| = 5000$, $\alpha = 0.8$ and $\lambda = 50/second$

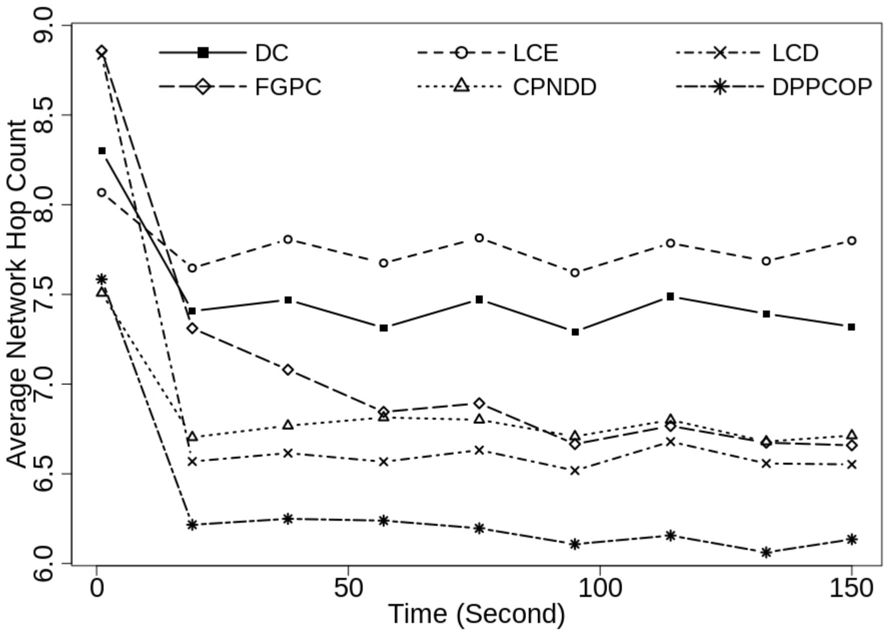


Figure 5.12: Average network hop-count with C_{size}=50, $|D| = 5000$, $\alpha = 1.1$ and $\lambda = 50/second$

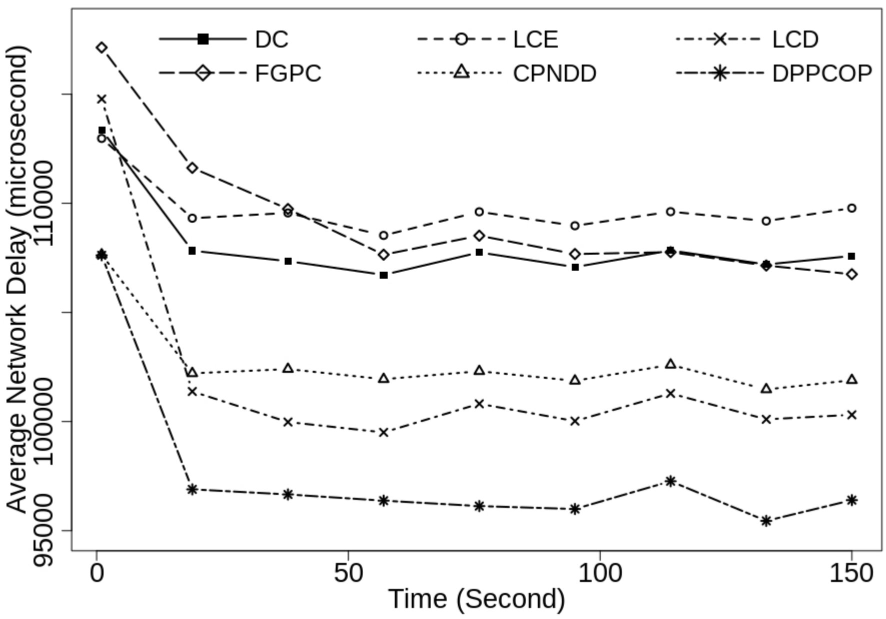


Figure 5.13: Average network delay (in microsecond) with C_{size}=50, $|D| = 5000$, $\alpha = 0.8$ and $\lambda = 50/second$

5.2.3 Effect on average network delay (DPPCOP caching scheme)

Fig. 5.13 and Fig. 5.14 plots the results of average network delay for the caching schemes under the different values of α (0.8 and 1.1). During simulations, it has been noticed that analogous to previous results, the network delay decreases when α increases. In low popularity distributions, all the caching schemes take a longer time to retrieve the contents. This is explained in section 2.1.3 for the Interest messages. Hence, most of the Interest messages cross a shorter path to access the matching content with an increase in α. For instance, Fig. 5.13 shows that the proposed DPPCOP caching strategy reduces the average network delay up to 12.2% from peer caching strategies.

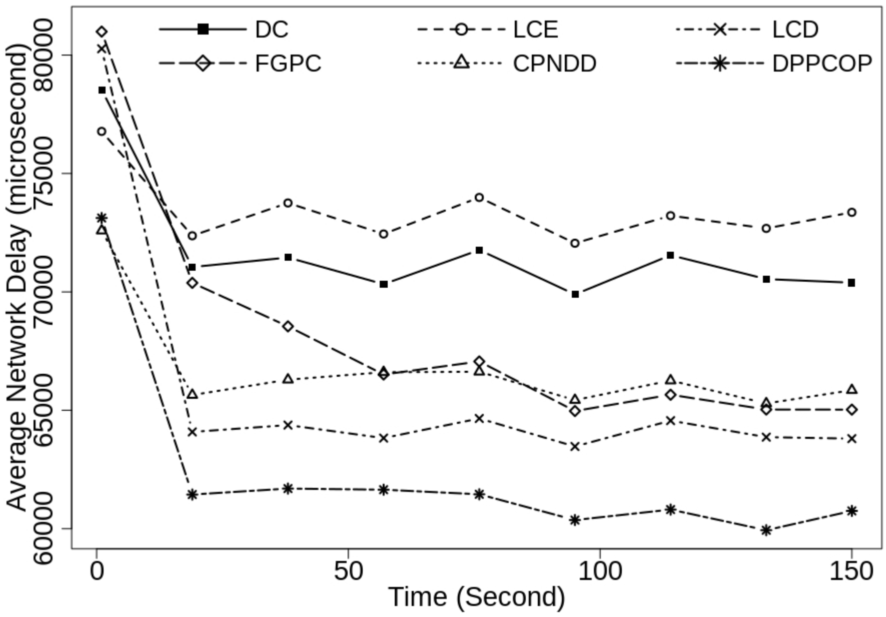


Figure 5.14: Average network delay (in microsecond) with C_{size}=50, $|D| = 5000$, $\alpha = 1.1$ and $\lambda = 50/second$

5.2.4 Effect on average network traffic (DPPCOP caching scheme)

The average network traffic reduces when the contents are accessed from the nearby in-network routers. Using network partitioning, the proposed scheme decreases cache redundancy, and thus, it caches diverse contents within partitions. Due to this, a large subset of popular contents is cached within each partition. Thus, a lesser number of Interest messages reaches the server for content retrieval and the requests are handled within partitions with increased probability.

Fig. 5.15 and Fig. 5.16 shows the average network traffic in terms of load on the network links in per unit time (KB/s) for different content popularity distributions ($\alpha = 0.8/1.1$). The decrease in the network traffic leads to lesser network congestion and improved QoS for the requesters. As shown in Fig. 5.15 when $\alpha = 0.8$, the proposed caching scheme outperforms the LCE, LCD, DC-based, FGPC, and CPNDD strategies by reducing the network traffic by 8.3%, 3.3%, 8.4%, 7.7%, and 5.1% respectively. A

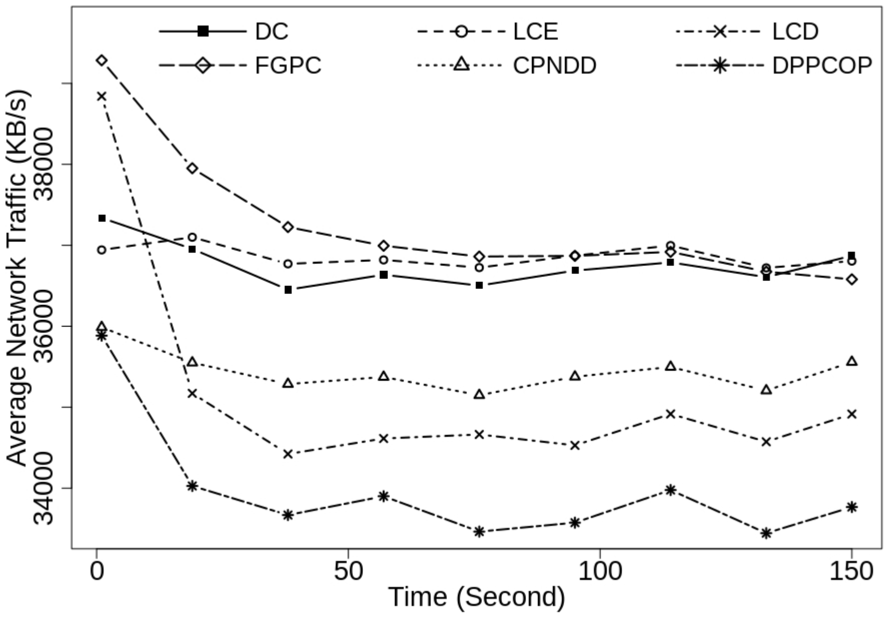


Figure 5.15: Average network traffic (in KB/second) with C_{size}=50, $|D| = 5000$, $\alpha = 0.8$ and $\lambda = 50/second$

similar reduction in network traffic is observed when α is increased to 1.1 because the caching probability is improved for the popular contents using the proposed heuristics. The simulation results, obtained with $\alpha = 1.1$ while keeping the remaining parameters unchanged, are shown in Fig. 5.16.

5.2.5 Discussion

As the in-network content caching shows significant improvement in the QoS for the end-user devices, it makes the caching feature suitable for the Content-Centric Networking based IoT environments [117] such as smart cities, smart healthcare systems, and smart transportation systems, etc. For example; a new content has been launched and a flash crowd occurred for this content in a smart city, then without in-network caching capability, only the server will provide the requested content to all users. In the flash crowds, a large number of end-users request for specific content. It would create congestion in the network and causes degraded QoS for the end-users. To minimize the load of the flash crowd, the proposed in-network caching

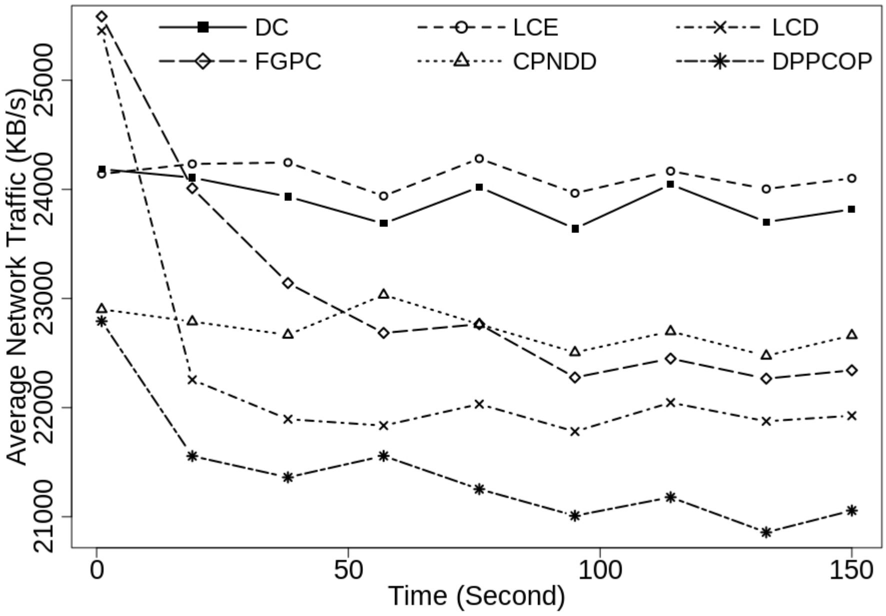


Figure 5.16: Average network traffic (in KB/second) with C_{size}=50, $|D| = 5000$, $\alpha = 1.1$ and $\lambda = 50/second$

scheme reduces the network traffic on the original server by caching the popular contents on suitable intermediate routers/gateways/access points. Similarly, in smart transportation systems, the vehicles can receive the traffic information from the nearby RSUs (Roadside Units) (having caching capabilities) instead of repeatedly fetching the same information from the servers.

Analogous to smart cities and transportations, the proposed scheme can also be deployed in the WSN [79]. In WSN, different applications can access the information from the sensors. In this scenario, instead of accessing the identical content from the sensors for each request, the content can be placed in the intermediate routers/gateways to forward the requested content towards applications. This would reduce the load from the sensor nodes and increase the "alive" duration for the power-constraint sensor devices. For efficient caching decisions in the routers, the network partitioning and content popularity along with the distance traversed by the Content message are taken into consideration, so that popular contents are cached near to the end-users with high probability. Due to this, the cache hit ratio has been increased and the contents are accessed with lesser delay and network traffic. This makes the proposed scheme suitable for the CCN-based IoT applications and wireless sensor networks.

5.3 Summary

This chapter evaluated the performance of the proposed PDC and DP-PCOP caching schemes. The performance of these schemes is compared with the traditional and recent caching strategies on identical simulation setups. The PDC and DPPCOP schemes are compared on performance metrics like cache hit ratio, average network hop count, average network delay, and network traffic parameters. The simulation results are presented and it is observed that the proposed schemes improve the network performance and QoS for the requesters as compared to the existing caching schemes. In the next chapter, the statistical validation of proposed and traditional caching schemes is presented.

Chapter 6

Statistical validations

In this section, the statistical validation of the proposed PDC and DP-PCOP schemes have been performed on the cache hit-ratio, hop-count, delay, and network traffic parameters. To analyze, whether the proposed schemes show significant performance improvement over existing schemes or not, the two-tailed T-test with unequal variance has been performed on the simulation results. It compares the performance of the proposed caching strategies scheme with the existing schemes.

6.1 Two-tailed T-Test with unequal variance

The two-tailed T-test is used to test the hypothesis that two results have equal mean values [118]. This statistical test is more reliable and accurate when the results have unequal variances and/or have an unequal number of sample results [119]. The T-test with unequal variance is also known as "unpaired" or "independent sample" T-tests because the statistical units of the two results are non-overlapping. The statistic t in the T-test is computed using the following equation:

$$t = \frac{(\bar{X_1} - \bar{X_2})}{\sqrt{s^2_{\bar{X}_1} + s^2_{\bar{X}_2}}} \tag{6.1}$$

In Equation 6.1, the $\bar{X_1}$ and $\bar{X_2}$ are the sample mean of the two results. For a given standard deviation and number of results, the standard errors $s_{\bar{X}_1}$ and $s_{\bar{X}_2}$ are computed using Equation 6.2.

$$s_{\bar{X}_i} = \frac{s_i}{\sqrt{N_i}} \tag{6.2}$$

Here, N_i represents the number of results obtained from a caching scheme during execution.

6.2 Null Hypothesis and alternative hypothesis

In a two-tailed T-test, the null hypothesis needs to be carefully defined. The following null hypothesis has been tested on the simulation results that are obtained for the caching strategies, "There is no significant difference between the hit-ratio, delay, hop-count, and network traffic results of the proposed caching solutions and the existing schemes". In the statistical analysis, a P-value defines the probability that the null hypothesis is true or false. A P-value is computed using the simulation results. If the P-value is greater than 0.05, then it indicates that there is more than 5% probability that the null hypothesis is true and the results are not significantly different from each other. It concludes that there is no evidence that the results of caching schemes are different even though they have different standard deviations. If, P-value ≤ 0.05, then the null hypothesis is rejected [120]. When the null hypothesis is rejected, then the alternative hypothesis becomes true i.e. "there is a significant difference between the results which are obtained for different caching strategies".

6.3 Statistical validation of the performance of PDC caching scheme

The statistical validation of the proposed PDC caching scheme is performed using the above-mentioned two-tailed T-test. The results of the existing peer caching schemes are compared with the PDC caching scheme on the hit ratio, average network hop count, delay, and network traffic metrics. Table 6.1 shows the obtained P-value for the hit ratio and the hop count parameters. The results demonstrate that the obtained P-value is significantly lesser than 0.05 and hence, the null hypothesis is rejected which indicates that the obtained results for the PDC scheme are significantly different than the existing caching schemes.

Table 6.1: Statistical validations of the performance of PDC over existing peer scheme: Hit ratio and average network hop count

Existing scheme	$(C_{size}(R_i))$	P-value (Hit-Ratio)	P-value (Average Hop-Count)
LCE	50	0.01 (3.26×10^{-155})	0.01 (7×10^{-160})
DC	50	0.01(1.7×10^{-128})	0.01 (1.2×10^{-117})
FGPC	50	0.01(7.8×10^{-100})	0.01 (2.12×10^{-79})
LCE	100	0.01(4.3×10^{-141})	0.01 (1.9×10^{-148})
DC	100	0.01(1.16×10^{-91})	0.01 (1.03×10^{-88})
FGPC	100	0.01(5.34×10^{-71})	0.01 (7.76×10^{-51})

Analogous to the results obtained for hit ratio and hop count metrics, the proposed PDC scheme shows significantly improved performance in terms of average network delay and traffic metrics as illustrated in Table 6.2. Hence, the content caching heuristics used in the proposed PDC scheme improves the network performance and QoS for the requesters and the results are also statistically validated.

Table 6.2: Statistical validations of the performance of DPPCOP over existing peer scheme: Average network delay and network traffic

Existing scheme	$(C_{size}(R_i))$	P-value (Average Network Delay)	P-value (Network Traffic)
LCE	50	0.01 (2.9×10^{-119})	0.01 (9.36×10^{-40})
DC	50	0.01 (1.59×10^{-87})	0.01 (3.94×10^{-34})
FGPC	50	0.01 (4.61×10^{-84})	0.01 (9.86×10^{-74})
LCE	100	0.01 (8.6×10^{-112})	0.01 ($1.0a7 \times 10^{-59}$)
DC	100	0.01 (6.39×10^{-70})	0.01 (1.6×10^{-50})
FGPC	100	0.01 (7.96×10^{-54})	0.01 (1.68×10^{-65})

6.4 Statistical validation of the performance of DPPCOP caching scheme

Table 6.3 and Table 6.4 show the computed P-value of the two-tailed T-Test that are obtained after the comparison of performance parameters of the proposed DPPCOP scheme with peer schemes. As the computed P-value is significantly lesser than the significance level (0.05) in the T-test, the null hypothesis has been rejected. Therefore, statistically, the performance achieved by the DPPCOP caching scheme is significantly different than the peer schemes. Hence, the performance of the DPPCOP is significantly higher (with $> 95\%$ confidence) and statistically validated.

6.5 Summary

This chapter discussed the significance of statistical analysis. The proposed PDC and DPPCOP caching strategies are statistically validated on

Table 6.3: Statistical validations of the performance of DPPCOP over existing peer scheme: Hit ratio and average network hop count

Existing scheme	(α)	P-value (Hit-Ratio)	P-value (Average Hop-Count)
LCE	0.8	0.01 (7.01×10^{-159})	0.01 (1.49×10^{-192})
DC	0.8	0.01(5.57×10^{-154})	0.01 (1.19×10^{-184})
FGPC	0.8	0.01(6.97×10^{-172})	0.01 (2.81×10^{-132})
LCD	0.8	0.01(1.51×10^{-78})	0.01 (1.48×10^{-49})
CPNDD	0.8	0.01(4.95×10^{-85})	0.01 (1.24×10^{-118})
LCE	1.1	0.01(5.39×10^{-168})	0.01 (4.18×10^{-198})
DC	1.1	0.01(1.76×10^{-165})	0.01 (3.26×10^{-190})
FGPC	1.1	0.01(3.31×10^{-72})	0.01 (1.07×10^{-44})
LCD	1.1	0.01(5.02×10^{-48})	0.01 (9.99×10^{-27})
CPNDD	1.1	0.01 (1.53×10^{-95})	0.01 (1.39×10^{-109})

Table 6.4: Statistical validations of the performance of DPPCOP over existing peer scheme: Average network delay and network traffic

Existing scheme	(α)	P-value (Average Network Delay)	P-value (Network Traffic)
LCE	0.8	0.01 (7.25×10^{-178})	0.01 (2.83×10^{-182})
DC	0.8	0.01 (1.74×10^{-172})	0.01 (8.86×10^{-178})
FGPC	0.8	0.01 (2.02×10^{-143})	0.01 (8.43×10^{-159})
LCD	0.8	0.01 (1.09×10^{-44})	0.01 (8.76×10^{-46})
CPNDD	0.8	0.01 (1.58×10^{-111})	0.01 (3.13×10^{-113})
LCE	1.1	0.01 (2.67×10^{-189})	0.01 (7.39×10^{-205})
DC	1.1	0.01 (2.45×10^{-180})	0.01 (7.71×10^{-195})
FGPC	1.1	0.01 (2.21×10^{-48})	0.01 (1.31×10^{-58})
LCD	1.1	0.01 (5.52×10^{-27})	0.01 (1.74×10^{-32})
CPNDD	1.1	0.01 (4.66×10^{-108})	0.01 (3.94×10^{-126})

different performance metrics under a wide variety of simulation configurations. In the next chapter, the conclusion and future scope of the thesis are presented.

Chapter 7

Conclusion and future research

The content caching characteristic of Content-Centric Networking has raised new challenges for efficient utilization of the available resources. Towards this, two novel content caching schemes are proposed in this work to improve network performance. Section 7.1 discusses the contributions of this thesis work. A brief on the future scope of the thesis is presented in Section 7.2.

7.1 Contributions

The major contributions of the thesis are as follows,

I Initially, a content popularity and distance parameters based caching scheme called PDC is proposed for the content caching decisions. To determine the content popularity with accuracy and reliability, a novel Popularity Window is proposed and the size of Popularity Window is computed using the content catalog size. Using the applied heuristics, the PDC caching scheme places the popular contents near the requesters to improve their QoS. The PDC scheme takes autonomous caching decisions without additional communication overhead. The simulation results show that the PDC scheme improves network performance as compared to the existing caching schemes under different caching capacities ($1\% - 2\%$ of the content catalog size).

II To control excessive caching operations and a further improvement in the performance of the PDC caching scheme, a dynamic partitioning-based caching scheme (DPPCOP) is proposed. The DPPCOP scheme dynamically partitioned the network into a sufficiently "good" number of partitions using the "Elbow" method. During partitioning, the scheme considers the hop count and the bandwidth parameters. Within each partition, the routers collaborate with each other for content caching decisions. The intra-partition routers ensure that at most one copy of the content is cached within a partition during the content forwarding operation. This method reduces content caching operations and content redundancy in the network without a significant increase in the communication overhead. For content placement decisions, the DPPCOP scheme uses the content popularity and hop count parameters with the LRU cache replacement algorithm.

III Extensive simulations are performed on the Abilene network topology for different content popularity distributions. When Zipf exponent value $\alpha = 0.8$, the simulation results demonstrate the superiority of the DPPCOP scheme on the existing caching schemes. In this scenario, the DPPCOP scheme increases the hit ratio up to 4.8% and reduces the average network hop count, delay, and network traffic by 12.2%, 15.1%, and 8.4% respectively from the peer caching schemes. A similar performance gain is delivered by the DPPCOP scheme from peer competing strategies when $\alpha = 1.1$. Hence, it makes the scheme suitable for implementation in performance-oriented applications such as WSN, VANET, IoT (Internet-of-Things), and upcoming

6G network architectures.

7.2 Future Scope

The content caching design in CCN posses several advantages during the content distributions as compared to existing IP-based Internet architecture. However, CCN is a rapidly changing environment with dynamic demands, trends, and technologies, which may often nullify with time. Hence, in future, many research challenges might need to be addressed in the proposed content caching solutions. Various areas of potential future works in the context of the presented work in this thesis are as follows:

I For content popularity determination, the proposed schemes implemented a novel Dynamic Popularity Window in each network router that determines the content access probability based on the past requests. The scheme considers the content catalog size to compute the size of the Popularity Window. In future, the machine learning algorithms can be integrated with the Popularity Window to predict future content requests with more accuracy. The optimal size of the Popularity Window can be determined based on these algorithms to optimize computational delay.

II During network partitioning, the proposed network partitioning scheme stops the merging of partitions when WSS values form an "Elbow". However, if "Elbow" is not formed during the merging of the partitions then it would become difficult to get sufficiently "good" number of partitions. In those scenarios, the network partitioning can be performed using "average silhouette method" or "gap statistical method" etc.

III The simulations of the caching solutions are performed in the ndnSIM simulation environment with an assumption that network nodes do not have energy constraints. The current version of ndnSIM (i.e. ndnSIM 2.0) has also not incorporated the energy-related metrics in the performance evaluation attributes for CCN such as energy consumption in a node or in the entire network in per unit time. Hence, in future, the performance of the energy-related metrics can be further explored and compared using custom-built network simulators.

Bibliography

[1] Andrew S Tanenbaum and David J Wetherall. Computer networks fifth edition. In *Pearson Education, Inc.* Prentice Hall, 2011.

[2] Bimal Kumar Mishra and Navnit Jha. Seiqrs model for the transmission of malicious objects in computer network. *Applied Mathematical Modelling*, 34(3):710–715, 2010.

[3] Carl A Sunshine. A brief history of computer networking. In *Computer Network Architectures and Protocols*, pages 3–6. Springer, 1989.

[4] Michael E Porter and Michael; ilustraciones Gibbs. Strategy and the internet. 2001.

[5] Jean-Philippe Vasseur and Adam Dunkels. *Interconnecting smart objects with ip: The next internet*. Morgan Kaufmann, 2010.

[6] Robert La Lau. Domain name system (dns). In *Practical Internet Server Configuration*, pages 141–158. Springer, 2021.

[7] Mark Handley. Why the internet only just works. *BT Technology Journal*, 24(3):119–129, 2006.

[8] William Stallings. Mobile ip. *The Internet Protocol Journal*, 4(2):2–14, 2001.

[9] Mayoon Yaibuates and Roungsan Chaisricharoen. Starvation delayed dhcp service for enabling pool recovery. *Malaysian Journal of Computer Science*, pages 15–34, 2019.

[10] Ki-Sik Kong, Wonjun Lee, Youn-Hee Han, Myung-Ki Shin, and HeungRyeol You. Mobility management for all-ip mobile networks: mobile ipv6 vs. proxy mobile ipv6. *IEEE Wireless communications*, 15(2):36–45, 2008.

[11] Sally Floyd et al. Highspeed tcp for large congestion windows. 2003.

[12] Lisong Xu, Khaled Harfoush, and Injong Rhee. Binary increase congestion control (bic) for fast long-distance networks. In *IEEE INFOCOM 2004*, volume 4, pages 2514–2524. IEEE, 2004.

[13] Donna Ghosh, Venkatesh Sarangan, and Raj Acharya. Quality-of-service routing in ip networks. *IEEE Transactions on Multimedia*, 3(2):200–208, 2001.

[14] Tony Bourke. *Server load balancing.* "O'Reilly Media, Inc.", 2001.

[15] Dweepayan Mishra, Arzeena Khan, Rajeev Tiwari, and Shuchi Upadhay. Automated irrigation system-iot based approach. In *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pages 1–4. IEEE, 2018.

[16] Rajeev Tiwari, Hitesh Kumar Sharma, Shuchi Upadhyay, Shreyansh Sachan, and Ayush Sharma. Automated parking system-cloud and iot based.

[17] Etqad Khan, Dipesh Garg, Rajeev Tiwari, and Shuchi Upadhyay. Automated toll tax collection system using cloud database. In *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pages 1–5. IEEE, 2018.

[18] Jennifer Rexford and Constantine Dovrolis. Future internet architecture: clean-slate versus evolutionary research. *Communications of the ACM*, 53(9):36–40, 2010.

[19] David R Cheriton and Mark Gritter. Triad: A new next-generation internet architecture. 2000.

[20] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica. A data-oriented (and beyond) network architecture. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 181–192, 2007.

[21] Lixia Zhang, Deborah Estrin, Jeffrey Burke, Van Jacobson, James D Thornton, Diana K Smetters, Beichuan Zhang, Gene Tsudik, Dan Massey, Christos Papadopoulos, et al. Named data networking (ndn) project. *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 157:158, 2010.

[22] Muhammad Azfar Yaqub, Syed Hassan Ahmed, Safdar Hussain Bouk, and Dongkyun Kim. Information-centric networks (icn). In *Content-Centric Networks*, pages 19–33. Springer, 2016.

[23] Van Jacobson, Diana K Smetters, James D Thornton, Michael F Plass, Nicholas H Briggs, and Rebecca L Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 1–12, 2009.

[24] Rihab Jmal and Lamia Chaari Fourati. Content-centric networking management based on software defined networks: Survey. *IEEE*

Transactions on network and service management, 14(4):1128–1142, 2017.

[25] Mohamed Diallo, Serge Fdida, Vasilis Sourlas, Paris Flegkas, and Leandros Tassiulas. Leveraging caching for internet-scale content-based publish/subscribe networks. In *2011 IEEE International Conference on Communications (ICC)*, pages 1–5. IEEE, 2011.

[26] George Xylomenos, Christopher N Ververidis, Vasilios A Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V Katsaros, and George C Polyzos. A survey of information-centric networking research. *IEEE communications surveys & tutorials*, 16(2):1024–1049, 2013.

[27] Giovanna Carofiglio, Massimo Gallo, Luca Muscariello, and Diego Perino. Modeling data transfer in content-centric networking. In *2011 23rd International Teletraffic Congress (ITC)*, pages 111–118. IEEE, 2011.

[28] Bormann Adamson, C Bormann, M Handley, and J Macker. Multicast negative-acknowledgment (nack) building blocks, 2008.

[29] Sumit Kumar, Rajeev Tiwari, Mohammad S Obaidat, Neeraj Kumar, and Kuei-Fang Hsiao. CPNDD: Content placement approach in content centric networking. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2020.

[30] Rajeev Tiwari and Neeraj Kumar. Cooperative gateway cache invalidation scheme for internet-based vehicular ad hoc networks. *Wireless Personal Communications*, 85(4):1789–1814, 2015.

[31] Rajeev Tiwari and Neeraj Kumar. An adaptive cache invalidation technique for wireless environments. *Telecommunication Systems*, 62(1):149–165, 2016.

[32] Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Borje Ohlman. A survey of information-centric networking. *IEEE Communications Magazine*, 50(7):26–36, 2012.

[33] Ioannis Psaras, Richard G Clegg, Raul Landa, Wei Koong Chai, and George Pavlou. Modelling and evaluation of ccn-caching trees. In *International Conference on Research in Networking*, pages 78–91. Springer, 2011.

[34] Faria Khandaker, Sharief Oteafy, Hossam S Hassanein, and Hesham Farahat. A functional taxonomy of caching schemes: Towards guided designs in information-centric networks. *Computer Networks*, 165:106937, 2019.

[35] Dario Rossi and Giuseppe Rossini. Caching performance of content centric networks under multi-path routing (and more). *Relatório técnico, Telecom ParisTech*, 2011:1–6, 2011.

[36] Kideok Cho, Munyoung Lee, Kunwoo Park, Ted Taekyoung Kwon, Yanghee Choi, and Sangheon Pack. Wave: Popularity-based and collaborative in-network caching for content-oriented networks. In *2012 Proceedings IEEE INFOCOM Workshops*, pages 316–321. IEEE, 2012.

[37] Huan Yan, Deyun Gao, Wei Su, Chuan Heng Foh, Hongke Zhang, and Athanasios V Vasilakos. Caching strategy based on hierarchical cluster for named data networking. *IEEE Access*, 5:8433–8443, 2017.

[38] Sen Wang, Jun Bi, and Jianping Wu. Collaborative caching based on hash-routing for information-centric networking. *ACM SIGCOMM Computer Communication Review*, 43(4):535–536, 2013.

[39] Christine Fricker, Philippe Robert, James Roberts, and Nada Sbihi. Impact of traffic mix on caching performance in a content-centric network. In *2012 Proceedings IEEE INFOCOM Workshops*, pages 310–315. IEEE, 2012.

[40] Santosh Fatale, R Sri Prakash, and Sharayu Moharir. Caching policies for transient data. *IEEE Transactions on Communications*, 68(7):4411–4422, 2020.

[41] Ioannis Psaras, Wei Koong Chai, and George Pavlou. In-network cache management and resource allocation for information-centric networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(11):2920–2931, 2013.

[42] Liang Wang, Gareth Tyson, Jussi Kangasharju, and Jon Crowcroft. Milking the cache cow with fairness in mind. *IEEE/ACM Transactions on Networking*, 25(5):2686–2700, 2017.

[43] Jakob Pfender, Alvin Valera, and Winston KG Seah. Easy as abc: A lightweight centrality-based caching strategy for information-centric iot. In *Proceedings of the 6th ACM conference on information-centric networking*, pages 100–111, 2019.

[44] M Dakshayini and R Rashmi. Hierarchical cooperative-peers fog caching strategy to improve the performance of content centric networking. *Multimedia Tools and Applications*, 80(13):20345–20368, 2021.

[45] Nidhi Lal, Shishupal Kumar, Garima Kadian, and Vijay Kumar Chaurasiya. Caching methodologies in content centric networking (ccn): A survey. *Computer Science Review*, 31:39–50, 2019.

[46] Martin Dräxler and Holger Karl. Efficiency of on-path and off-path caching strategies in information centric networks. In *2012 IEEE International Conference on Green Computing and Communications*, pages 581–587. IEEE, 2012.

[47] Sen Wang, Jun Bi, Jianping Wu, and Athanasios V Vasilakos. Cphr: In-network caching for information-centric networking with partitioning and hash-routing. *IEEE/ACM transactions on networking*, 24(5):2742–2755, 2015.

[48] Andriana Ioannou and Stefan Weber. Towards on-path caching alternatives in information-centric networks. In *39th Annual IEEE Conference on Local Computer Networks*, pages 362–365. IEEE, 2014.

[49] Zhe Li and Gwendal Simon. Time-shifted tv in content centric networks: The case for cooperative in-network caching. In *2011 IEEE international conference on communications (ICC)*, pages 1–6. IEEE, 2011.

[50] Sumanta Saha, Andrey Lukyanenko, and Antti Ylä-Jääski. Cooperative caching through routing control in information-centric networks. In *2013 Proceedings IEEE INFOCOM*, pages 100–104. IEEE, 2013.

[51] HyunYong Lee and Akihiro Nakao. User-assisted in-network caching in information-centric networking. *Computer Networks*, 57(16):3142–3153, 2013.

[52] Jun Li, Hao Wu, Bin Liu, Zhaoxi Fang, Shaozhong Zhang, and Jiong Shi. Rbc-cc: Rbc-based cascade caching scheme for content-

centric networking. *Journal of Network and Systems Management*, 25(2):375–396, 2017.

[53] Jason Min Wang, Jun Zhang, and Brahim Bensaou. Intra-as cooperative caching for content-centric networks. In *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, pages 61–66, 2013.

[54] Vasilis Sourlas, Lazaros Gkatzikis, Paris Flegkas, and Leandros Tassiulas. Distributed cache management in information-centric networks. *IEEE Transactions on Network and Service Management*, 10(3):286–299, 2013.

[55] Somaya Arianfar, Pekka Nikander, and Jörg Ott. On content-centric router design and implications. In *Proceedings of the re-architecting the internet workshop*, pages 1–6, 2010.

[56] Ioannis Psaras, Wei Koong Chai, and George Pavlou. Probabilistic in-network caching for information-centric networks. In *Proceedings of the second edition of the ICN workshop on Information-centric networking*, pages 55–60, 2012.

[57] Dario Rossi and Giuseppe Rossini. On sizing ccn content stores by exploiting topological information. In *2012 Proceedings IEEE INFOCOM Workshops*, pages 280–285. IEEE, 2012.

[58] Wei Koong Chai, Diliang He, Ioannis Psaras, and George Pavlou. Cache less for more in information-centric networks (extended version). *Computer Communications*, 36(7):758–770, 2013.

[59] Vasilis Sourlas, Georgios S Paschos, Paris Flegkas, and Leandros Tassiulas. Caching in content-based publish/subscribe systems. In

GLOBECOM 2009-2009 IEEE Global Telecommunications Conference, pages 1–6. IEEE, 2009.

[60] Kumari Nidhi Lal and Anoj Kumar. A centrality-measures based caching scheme for content-centric networking (ccn). *Multimedia Tools and Applications*, 77(14):17625–17642, 2018.

[61] Nikolaos Laoutaris, Hao Che, and Ioannis Stavrakakis. The lcd interconnection of lru caches and its analysis. *Performance Evaluation*, 63(7):609–634, 2006.

[62] Zhongxing Ming, Mingwei Xu, and Dan Wang. Age-based cooperative caching in information-centric networks. In *2012 Proceedings IEEE INFOCOM Workshops*, pages 268–273. IEEE, 2012.

[63] Muhammad Ali Naeem, Shahrudin Awang Nor, Suhaidi Hassan, and Byung-Seo Kim. Compound popular content caching strategy in named data networking. *Electronics*, 8(7):771, 2019.

[64] Sheneela Naz, Rao Naveed Bin Rais, and Amir Qayyum. Multi-attribute caching: Towards efficient cache management in content-centric networks. In *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 630–633. IEEE, 2016.

[65] Fan-Hsun Tseng, Wei-Che Chien, Sheng-Jie Wang, Chin Feng Lai, and Han-Chieh Chao. A novel cache scheme based on content popularity and user locality for future internet. In *2018 27th Wireless and Optical Communication Conference (WOCC)*, pages 1–5. IEEE, 2018.

[66] Hsin-Te Wu, Hsin-Hung Cho, Sheng-Jie Wang, and Fan-Hsun Tseng. Intelligent data cache based on content popularity and user location

for content centric networks. *Human-centric Computing and Information Sciences*, 9(1):1–16, 2019.

[67] Faiza Qazi, Osman Khalid, Rao Naveed Bin Rais, Imran Ali Khan, et al. Optimal content caching in content-centric networks. *Wireless Communications and Mobile Computing*, 2019, 2019.

[68] Mingchuan Zhang, Ping Xie, Junlong Zhu, Qingtao Wu, Ruijuan Zheng, and Hongke Zhang. Ncpp-based caching and nur-based resource allocation for information-centric networking. *Journal of Ambient Intelligence and Humanized Computing*, 10(5):1739–1745, 2019.

[69] César Bernardini, Thomas Silverston, and Olivier Festor. Mpc: Popularity-based caching strategy for content centric networks. In *2013 IEEE international conference on communications (ICC)*, pages 3619–3623. IEEE, 2013.

[70] Sung-Hwa Lim, Young-Bae Ko, Gue-Hwan Jung, Jaehoon Kim, and Myeong-Wuk Jang. Inter-chunk popularity-based edge-first caching in content-centric networking. *IEEE Communications Letters*, 18(8):1331–1334, 2014.

[71] Long Chen, Hongbo Tang, Xingguo Luo, Yi Bai, and Zhen Zhang. Gain-aware caching scheme based on popularity monitoring in information-centric networking. *IEICE Transactions on Communications*, page 2015EBP3539, 2016.

[72] Andrea Detti, Lorenzo Bracciale, Pierpaolo Loreti, Giulio Rossi, and Nicola Blefari Melazzi. A cluster-based scalable router for information centric networks. *Computer networks*, 142:24–32, 2018.

[73] Kamrul Hasan and Seong-Ho Jeong. Efficient caching for data-driven iot applications and fast content delivery with low latency in icn. *Applied Sciences*, 9(22):4730, 2019.

[74] Bitan Banerjee, Adita Kulkarni, and Anand Seetharam. Greedy caching: An optimized content placement strategy for information-centric networks. *Computer Networks*, 140:78–91, 2018.

[75] Jing Ren, Wen Qi, Cedric Westphal, Jianping Wang, Kejie Lu, Shucheng Liu, and Sheng Wang. Magic: A distributed max-gain in-network caching strategy in information-centric networks. In *2014 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*, pages 470–475. IEEE, 2014.

[76] Mau Dung Ong, Min Chen, Tarik Taleb, Xiaofei Wang, and Victor CM Leung. Fgpc: fine-grained popularity-based caching design for content centric networking. In *Proceedings of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pages 295–302, 2014.

[77] Sung-Won Lee, Dabin Kim, Young-Bae Ko, Jae-Hoon Kim, and Myeong-Wuk Jang. Cache capacity-aware ccn: Selective caching and cache-aware routing. In *2013 IEEE Global Communications Conference (GLOBECOM)*, pages 2114–2119. IEEE, 2013.

[78] Martin Everett and Stephen P Borgatti. Ego network betweenness. *Social networks*, 27(1):31–38, 2005.

[79] Ghada Jaber and Rahim Kacimi. A collaborative caching strategy for content-centric enabled wireless sensor networks. *Computer Communications*, 159:60 – 70, 2020.

[80] Kamrul Hasan and Seong-Ho Jeong. A cluster-based content management framework for Information-Centric Networking. In *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 891–893. IEEE, 2018.

[81] Ikram Ud Din, Suhaidi Hassan, Ahmad Almogren, Farrukh Ayub, and Mohsen Guizani. Puc: Packet update caching for energy efficient iot-based information-centric networking. *Future Generation Computer Systems*, 2019.

[82] Chengming Li and Koji Okamura. Cluster-based in-networking caching for content-centric networking. *International Journal of Computer Science and Network Security (IJCSNS)*, 14(11):1, 2014.

[83] Sumit Kumar and Rajeev Tiwari. An efficient content placement scheme based on normalized node degree in content centric networking. *Cluster Computing*, pages 1–15, 2020.

[84] Taiji Chen, Hui Li, Jinchen An, Yunmin Wang, Jingting Sun, and Haibin Wang. The improved multi-attribute caching: an efficient cache strategy in ccn. In *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, pages 1007–1011. IEEE, 2017.

[85] Yuemei Xu, Yang Li, Tao Lin, Guoqiang Zhang, Zihou Wang, and Song Ci. A dominating-set-based collaborative caching with request routing in content centric networking. In *2013 IEEE International Conference on Communications (ICC)*, pages 3624–3628. IEEE, 2013.

[86] Mengjun Xie, Indra Widjaja, and Haining Wang. Enhancing cache robustness for content-centric networking. In *2012 Proceedings IEEE INFOCOM*, pages 2426–2434. IEEE, 2012.

[87] Anirban Mahanti, Derek Eager, and Carey Williamson. Temporal locality and its impact on web proxy cache performance. *Performance Evaluation*, 42(2-3):187–203, 2000.

[88] Siyang Shan, Chunyan Feng, Tiankui Zhang, and Jonathan Loo. Proactive caching placement for arbitrary topology with multi-hop forwarding in icn. *IEEE Access*, 7:149117–149131, 2019.

[89] Hani Salah and Thorsten Strufe. Comon: An architecture for coordinated caching and cache-aware routing in ccn. In *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pages 663–670. IEEE, 2015.

[90] Lorenzo Saino, Ioannis Psaras, and George Pavlou. Icarus: a caching simulator for information centric networking (icn). In *SimuTools*, volume 7, pages 66–75. ICST, 2014.

[91] Sumit Kumar, Rajeev Tiwari, Sergei Kozlov, and Joel JPC Rodrigues. Minimizing delay in content-centric networks using heuristics-based in-network caching. *Cluster Computing*, pages 1–15, 2021.

[92] Yanhua Li, Haiyong Xie, Yonggang Wen, and Zhi-Li Zhang. Coordinating in-network caching in content-centric networks: Model and analysis. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*, pages 62–72. IEEE, 2013.

[93] Huan Yan, Deyun Gao, and Wei Su. A hierarchical cluster-based caching for named data networking. In *2016 IEEE/CIC International*

Conference on Communications in China (ICCC), pages 1–6. IEEE, 2016.

[94] Yan Chen, Lili Qiu, Weiyu Chen, Luan Nguyen, and Randy H Katz. Efficient and adaptive web replication using content clustering. *IEEE Journal on Selected Areas in Communications*, 21(6):979–994, 2003.

[95] L Kaufman and P Rousseeuw. Clustering by means of medoids reports of the faculty of mathematics and informatics. *Faculty of Mathematics and Informatics*, 1987.

[96] Vasilis Sourlas, Ioannis Psaras, Lorenzo Saino, and George Pavlou. Efficient hash-routing and domain clustering techniques for information-centric networks. *Computer Networks*, 103:67–83, 2016.

[97] Farooq Aftab, Zhongshan Zhang, and Adeel Ahmad. Self-organization based clustering in manets using zone based group mobility. *IEEE Access*, 5:27464–27476, 2017.

[98] Prasad Krishna, Nitin H Vaidya, Mainak Chatterjee, and Dhiraj K Pradhan. A cluster-based approach for routing in dynamic networks. *ACM SIGCOMM computer communication review*, 27(2):49–64, 1997.

[99] Sumit Kumar and Rajeev Tiwari. Dynamic popularity window and distance-based efficient caching for fast content delivery applications in ccn. *Engineering Science and Technology, an International Journal*, 24(3):829–837, 2021.

[100] Alexander Afanasyev, Junxiao Shi, Beichuan Zhang, Lixia Zhang, Ilya Moiseenko, Yingdi Yu, Wentao Shang, Yi Huang, Jerald Paul Abraham, Steve DiBenedetto, et al. Nfd developer's guide. *Dept.*

Comput. Sci., Univ. California, Los Angeles, Los Angeles, CA, USA, Tech. Rep. NDN-0021, 2014.

[101] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. Web caching and zipf-like distributions: Evidence and implications. In *IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320)*, volume 1, pages 126–134. IEEE, 1999.

[102] Ruyan Wang, Zunwei Kan, Yaping Cui, Dapeng Wu, and Yan Zhen. Cooperative caching strategy with content request prediction in internet of vehicles. *IEEE Internet of Things Journal*, 8(11):8964–8975, 2021.

[103] Sumit Kumar and Rajeev Tiwari. Optimized content centric networking for future internet: dynamic popularity window based caching scheme. *Computer Networks*, 179:107434, 2020.

[104] Yuemei Xu, Song Ci, Yang Li, Tao Lin, and Gang Li. Design and evaluation of coordinated in-network caching model for content centric networking. *Computer Networks*, 110:266–283, 2016.

[105] Chunhui Yuan and Haitao Yang. Research on k-value selection method of k-means clustering algorithm. *J-Multidisciplinary Scientific Journal*, 2(2):226–235, 2019.

[106] Congming Shi, Bingtao Wei, Shoulin Wei, Wen Wang, Hai Liu, and Jialei Liu. A quantitative discriminant method of elbow point for the optimal number of clusters in clustering algorithm. *EURASIP Journal on Wireless Communications and Networking*, 2021(1):1–16, 2021.

[107] Purnima Bholowalia and Arvind Kumar. Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9), 2014.

[108] Thi-Bich-Hanh Dao, Khanh-Chuong Duong, and Christel Vrain. Constrained minimum sum of squares clustering by constraint programming. In *International Conference on Principles and Practice of Constraint Programming*, pages 557–573. Springer, 2015.

[109] Danny Matthew Saputra, Daniel Saputra, and Liniyanti D Oswari. Effect of distance metrics in determining k-value in k-means clustering using elbow and silhouette method. In *Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019)*, pages 341–346. Atlantis Press, 2020.

[110] David Alderson, Lun Li, Walter Willinger, and John C Doyle. Understanding internet topology: principles, models, and validation. *IEEE/ACM Transactions on networking*, 13(6):1205–1218, 2005.

[111] Yanhua Li, Haiyong Xie, Yonggang Wen, Chi-Yin Chow, and Zhi-Li Zhang. How much to coordinate? optimizing in-network caching in content-centric networks. *IEEE Transactions on network and service management*, 12(3):420–434, 2015.

[112] Alexander Afanasyev, Ilya Moiseenko, Lixia Zhang, et al. ndnsim: Ndn simulator for ns-3. 2012.

[113] Spyridon Mastorakis, Alexander Afanasyev, Ilya Moiseenko, and Lixia Zhang. ndnsim 2: An updated ndn simulator for ns-3. *NDN, Technical Report NDN-0028, Revision 2*, 2016.

[114] Raffaele Chiocchetti, Dario Rossi, and Giuseppe Rossini. ccnsim: An highly scalable ccn simulator. In *2013 IEEE International Conference on Communications (ICC)*, pages 2309–2314. IEEE, 2013.

[115] Andras Varga. Omnet++. In *Modeling and tools for network simulation*, pages 35–59. Springer, 2010.

[116] César Bernardini, Thomas Silverston, and Olivier Festor. A comparison of caching strategies for content centric networking. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2015.

[117] Sobia Arshad, Muhammad Awais Azam, Mubashir Husain Rehmani, and Jonathan Loo. Recent advances in information-centric networking-based Internet of Things (icn-iot). *IEEE Internet of Things Journal*, 6(2):2128–2158, 2018.

[118] Graeme D Ruxton. The unequal variance t-test is an underused alternative to student's t-test and the mann–whitney u test. *Behavioral Ecology*, 17(4):688–690, 2006.

[119] Ben Derrick, Deirdre Toher, and Paul White. Why welch's test is type i error robust. *The Quantitative Methods in Psychology*, 12(1), 2016.

[120] Charles Zaiontz. Real statistics using excel. *Retrieved*, 10:2015, 2015.

Skt

ORIGINALITY REPORT

9%	3%	8%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Sumit Kumar, Rajeev Tiwari. "Optimized content centric networking for future internet: Dynamic popularity window based caching scheme", Computer Networks, 2020 Publication	2%
2	"Proceedings of Second Doctoral Symposium on Computational Intelligence", Springer Science and Business Media LLC, 2022 Publication	1%
3	Sumit Kumar, Rajeev Tiwari. "An efficient content placement scheme based on normalized node degree in content centric networking", Cluster Computing, 2020 Publication	1%
4	www.mdpi.com Internet Source	<1%
5	doaj.org Internet Source	<1%
6	Faria Khandaker, Sharief Oteafy, Hossam S. Hassanein, Hesham Farahat. "A functional taxonomy of caching schemes: Towards	<1%