COMPOUND POPULAR CONTENT CACHING STRATEGY TO ENHANCE THE CACHE MANAGEMENT PERFORMANCE IN NAMED DATA NETWORKING

MUHAMMAD ALI NAEEM

DOCTOR OF PHILOSOPHY
UNIVERSITI UTARA MALAYSIA
2020

Awang Had Salleh
Graduate School
of Arts And Sciences

Universiti Utara Malaysia

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

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

MUHAMMAD ALI NAEEM

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

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

"COMPOUND POPULAR CONTENT CACHING STRATEGY TO ENHANCE THE CACHE MANAGEMENT PERFORMANCE IN NAMED DATA NETWORKING"

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

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : ***12 Disember 2019.***
That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on:
December 12, 2019.

Pengerusi Viva: *(Chairman for VIVA)*	**Prof. Dr. Huda Hj Ibrahim**	Tandatangan *(Signature)*
Pemeriksa Luar: *(External Examiner)*	**Prof. Dr. Mohamed Othman**	Tandatangan *(Signature)*
Pemeriksa Dalam: *(Internal Examiner)*	**Dr. Amran Ahmad**	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	**Dr. Shahrudin Awang Nor**	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	**Prof. Dr. Suhaidi Hassan**	Tandatangan *(Signature)*

Tarikh:
(Date) ***December 12, 2019***

Permission to Use

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

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

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok

Abstrak

Rangkaian Data Dinamakan (NDN) adalah paradigma penyelidikan yang terkemuka untuk seni bina Internet masa depan. Ia bertujuan untuk membangunkan pendekatan penyebaran data yang cekap dengan melaksanakan penyimpanan cache dalam nod rangkaian. Pengagregatan adalah salah satu daripada seni bina NDN yang paling menonjol yang dapat meningkatkan struktur Internet dengan ketara. Ia dapat mengurangkan limpahan lalu lintas data global dengan menyediakan cache. Kajian ini membentangkan tinjauan komprehensif mengenai strategi pengagregatan berasaskan negara NDN yang bertujuan untuk menangani isu-isu pengagregatan, dengan tumpuan tertentu untuk meminimumkan kebolehlaksanaan cache dengan nisbah kepelbagaian yang dipertingkatkan dan meningkatkan kebolehcapaian kandungan dalam cache dengan laluan regangan pendek. Kajian ini mencadangkan strategi pengurusan cache baru yang dinamakan sebagai Strategi Pengagregatan Kandungan Popular Majumk (CPCCS). Untuk melaksanakan kajian yang dicadangkan, Metodologi Penyelidikan Reka Bentuk telah diaplikasikan dan CPCCS dikaji secara menyeluruh dan relatif dengan strategi pengagregatan berasaskan NDN yang lain, seperti Pengagregatan In-network MAX-Gain, strategi pengagregatan popular popular WAVE, Pengagregatan Probabilistic berasaskan Hop, Leaf Popular Down, Cache Paling Popular, dan Pengagregatan Sedar Kapasiti Cache Mengetahui dalam persekitaran simulasi. CPCCS telah dibangunkan dengan menggabungkan dua mekanisme yang dinamakan sebagai Pemilihan Kandungan Populer Kompaun dan Pengagregatan Kandungan Popular Majmuk untuk membezakan kandungan berkenaan frekuensi Minat mereka dan untuk mencari kedudukan pengagregatan yang terbaik. Keputusan menunjukkan bahawa CPCCS memberikan prestasi yang lebih baik dari segi Kepelbagaian sebanyak 34%, cache hit nisbah sebanyak 14%, redundansi sebanyak 44%, dan Regangkan sebanyak 46%. Hasilnya menunjukkan bahawa CPCCS telah mencapai kepelbagaian kandungan yang dipertingkatkan, nisbah kena cache, redundansi kandungan, dan regangan yang berkaitan dengan saiz cache yang berbeza (1GB hingga 10GB) daripada strategi Pengagregatan lain. Oleh itu, CPCCS mempunyai pengaruh penting untuk digunakan pada masa depan untuk teknologi baru yang berasaskan NDN seperti Internet of Things, kabus dan pengkomputeran tepi.

Kata kunci: Menunggu jadual minat, Masa sejak lahir, Penghala kandungan, Rangkaian penghantaran kandungan, Kajian preskriptif.

Abstract

Named Data Networking (NDN) is a leading research paradigm for the future Internet architecture. The NDN offers in-network cache which is the most beneficial feature to reduce the difficulties of the location-based Internet paradigm. The objective of cache is to achieve a scalable, effective, and consistent distribution of information. However, the main issue which NDN facing is the selection of appropriate router during the content's transmission that can disrupt the overall network performance. The reason is that how each router takes a decision to the cache which content needs to cache at what location that can enhance the complete caching performance. Therefore, several cache management strategies have been developed. Still, it is not clear which caching strategy is the most ideal for each situation. This study proposes a new cache management strategy named as Compound Popular Content Caching Strategy (CPCCS) to minimize cache redundancy with enhanced diversity ratio and improving the accessibility of cached content by providing short stretch paths. The CPCCS was developed by combining two mechanisms named as Compound Popular Content Selection (CPCS) and Compound Popular Content Caching (CPCC) to differentiate the contents regarding their Interest frequencies using dynamic threshold and to find the best possible caching positions respectively. CPCCS is compared with other NDN-based caching strategies, such as Max-Gain In-network Caching, WAVE popularity-based caching strategy, Hop-based Probabilistic Caching, Leaf Popular Down, Most Popular Cache, and Cache Capacity Aware Caching in a simulation environment. The results show that the CPCCS performs better in which the diversity and cache hit ratio are increased by 34% and 14% respectively. In addition, the redundancy and path stretch are decreased by 44% and 46% respectively. The outcomes showed that the CPCCS have achieved enhanced caching performance with respect to different cache size (1GB to 10GB) and simulation parameters than other caching strategies. Thus, CPCCS can be applicable in future for the NDN-based emerging technologies such as Internet of Things, fog and edge computing.

Keywords: Pending Interest Table, Cache redundancy, Content diversity, Stretch path, Content Store.

Declaration Associated with This Thesis

[1] Muhammad Ali Naeem, Shahrudin Awang Nor. A survey of content placement strategies for content-centric networking. In AIP Conference Proceedings 2016 Aug 12 (Vol. 1761, No. 1, p. 020078). AIP Publishing. ISI and Scopus Indexed.

[2] Muhammad Ali Naeem, Shahrudin Awang Nor, Suhaidi Hassan. "Information Centric Progressive Content Placement Cache for Multimedia data contents "Journal of Advanced Research in Dynamical and Control Systems 10 (2018), Page 1291-1301. Scopus Indexed.

[3] Muhammad Ali Naeem, Rashid Ali, Byung-Seo Kim, Shahrudin Awang Nor, and Suhaidi Hassan. "A periodic caching strategy solution for the smart city in information-centric Internet of Things." *Sustainability* 10, no. 7 (2018): 2576. ISI and Scopus Indexed.

[4] Muhammad Ali Naeem, Shahrudin Awang Nor, Suhaidi Hassan, and Byung-Seo Kim. "Performances of probabilistic caching strategies in content centric networking." *IEEE Access* 6 (2018): 58807-58825. Rank Q1, ISI and Scopus Indexed.

[5] Muhammad Ali Naeem, Shahrudin Awang Nor, Suhaidi Hassan (2018). "IP-Internet data Dissemination Challenges and Future Research Directions" Journal of Advanced Research in Dynamical and Control Systems. 10 (2018), Page: 1596-1606. Scopus Indexed.

[6] Muhammad Ali Naeem, Shahrudin Awang Nor, Suhaidi Hassan. "A Survey on Probabilistic Caching Mechanisms in Content Centric Networking" Journal of Advanced Research in Dynamical and Control Systems. 10 (2018). Pages: 1309-1321. Scopus Indexed.

[7] Muhammad Ali Naeem, Shahrudin Awang Nor, Suhaidi Hassan. "A Review on Content Distribution through Caching in Information Centric Networking" Journal of Advanced Research in Dynamical and Control Systems. 11 (2019). Pages: 213-220 Scopus Indexed.

[8] Muhammad Ali Naeem, Shahrudin Awang Nor, Suhaidi Hassan, and Byung-Seo Kim. "Compound Popular Content Caching Strategy in Named Data Networking." *Electronics* 8, no. 7 (2019): 771. Rank Q1, ISI and Scopus Indexed.

[9] Muhammad Ali Naeem, Shahrudin Awang Nor, Senior Member, IEEE, Suhaidi Hassan, Senior Member, IEEE, Byung-Seo Kim, Senior Member, IEEE, "Prospective Content Placement Strategy for Multimedia Data Content in Named Data Networking" IEEE System Journal Rank Q1, ISI and Scopus Indexed.

[10] Muhammad Ali Naeem, Shahrudin Awang Nor and Suhaidi Hassan Architectures in Information Centric networking: a critical review. Advances in Intelligent Systems and Computing book series, Computer Science (miscellaneous) (Q3). Book Chapter (Springer Book Series) ISSN: 2194-5357. 0.56. (**Indexing: The books of this series are submitted to ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springerlink).

[11] Hussaini, Muktar, Muhammad Ali Naeem, Byung-Seo Kim, and Inusa Sani Maijama'a. "Efficient Producer Mobility Management Model in Information-Centric Networking." *IEEE Access* 7 (2019): 42032-42051. (*IEEE Access*2169-3536, 2019).

Acknowledgement

All praise and glory be to Allah for granting me health, strength and knowledge to attain this stage of my life journey. The favors and mercy of preserving me are undeniable.

My profound and sincere appreciation goes to my Supervisor, Dr. Shahrudin Bin Awang Nor and Professor Dr. Suhaidi B Hassan for granting me the space and scholarly guidance through this wonderful journey. It would have been almost impossible without you. Your professionalism and daily inspection on the work and my well-being was a key. Working in this area with you has broadened my understanding and improved my thinking; I say a very big thank you and may GOD increase you in wisdom, strength, health and wealth.

My special acknowledgment is to my superior Supervisor, Professor. Dr. Suhaidi B Hassan. Your guidance is enormous that words are short to describe. I appreciate the guidance and advices through presentation and objective criticism that binds us to the Laboratory. I want to thank you for providing us with the benevolent and most conducive research environment.

To the esteem team of the dedicated thesis reviewers, the Chairman, Prof. Dr. Hud Hj Ibrahim and the examiners Dr. Amran Ahmad and Prof. Dr. Mohamed Othman. Your professional suggestions and recommendations were golden in adding value to the thesis. My out most appreciation goes to Dr. Amran Ahmad and Prof. Dr. Mohamed Othman who critically guided me through constructive inputs; without your sacrifices of reading and grading my work, the worthiness would not be there; I pray to God to increase you in knowledge health and wealth.

My unreserved gratitude goes to my parents, brothers, sisters, and especially to my wife; your prayers, supports, love and understanding kept me going. The strength of you all motivated my journey to this length. You all showered your motivation to me that I could do it. I pray that Allah almighty continue to strengthen our bonds and bless the entire household.

I want to specifically appreciate the efforts of the entire InterNetWorks Research Laboratory, Universiti Utara Malaysia team. You are my family forever and I shall live to be honored to have ever worked with you. Your motivation and supports kept me going even in the most difficult times of my study.

At the end, I would also like to thank all the staff of the School of Computing (SOC), Universiti Utara Malaysia (UUM) for their invaluable helps throughout my study.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ARPANET	-	Advanced Research Projects Agency Network
CCN	-	Content-Centric Network
CPCCS	-	Compound Popular Content Caching Strategy
CDN	-	Content Delivery Network
CR	-	Content Router
CS	-	Content Store
CT	-	Comparison Table
DNS	-	Domain Name System
DONA	-	Data-Oriented Network architecture
ICN	-	Information-Centric Network
IP	-	Internet Protocol
LCD	-	Leave Copy Down
LCE	-	Leave Copy Everywhere
LRU	-	Least Recently Used
MCD	-	Move Copy Down
MPC	-	Most Popular Content
NDN	-	Named Data Network
NDO	-	Named Data Object
NetInf	-	Network of Information
PIT	-	Pending Interest Table
PS	-	Prescriptive Study
PSIRP	-	Publish-Subscribe Internet Routing Paradigm

PT	-	Popularity Table
PURSUIT	-	Publisher Subscriber Internet Technology
RC	-	Research Clarification
TSB	-	Time Since Birth
TSI	-	Time Since Inception
VoD	-	Video on Demand
VNI	-	Visual Networking Index
WWW	-	World Wide Web (WWW)
WAN	-	Wide Area Network

CHAPTER ONE
INTRODUCTION

1.1 Background

The current Internet was basically developed to share distant computer resources but due to the unavailability of the required resources and security threats, this has not so far been materialized. And only a little number of people had access to it. The basic Internet has been changing with the incremental alteration in the architecture for providing all the services that currently the Internet offers. The initial Internet prototype was developed in 1967, known as the Advanced Research Projects Agency Network (ARPANET), connected only one specific type of computers/microcomputers [1]. Later on, different sorts of systems, for example, satellite and portable radio systems, were interconnected through ARPANET. According to the usage increment of the Internet, the TCP/IP stack was adopted by the ARPANET in 1983, to handle the increasing Internet connections [2]. In 1983, the Domain Name System (DNS) was made to lessen the multifaceted nature to discover these kinds of networks [3]. Of late, the development of the World Wide Web (WWW) allowed the Internet to achieve 600 million clients in late 2000 [4], and this number has continued developing from that point forward.

By the people awareness, the requirements were increasing for the Internet and the evolution of the Internet architecture did not follow the growing demands of Internet users and communicating data [5]. For example, the existing correspondence worldview of the Internet presents numerous restrictions. To begin with, Internet design can offer numerous duplicates of objects, yet these duplicates are not connected together. Keeping in mind the end goal to share assets of the system, extra overlay instruments have been proposed over the TCP/IP stack [6]. Reliable and fast object exchange requires application particular systems, for example, Content Delivery Networks (CDN) or Peer to Peer (P2P) administrations [7]. Secondly, security is accomplished by outsider applications and administrations. Trust in receiving objects is difficult to accomplish and the majority of the associations depend on deceitful locations. Third, the reality of receiving objects is strongly related to their location. Each Internet packet is tended to taking after source and destination addresses. Every time when an interest packet is sent for object accessing, it needs addresses. These requests are firmly connected to an address, flagging a specific location of the objects, when clients could not care about the location but get the object as fast as could be expected [8]. The direct manner to resolve this problem is to replace where with what. Host-to-host correspondence was a reflection to take care of the issues of the 1960s. So the researchers contend that architecture in view of named objects is better for the communication of this time [9].

In light of this reason, Information-Centric Networks (ICNs) was proposed in 2009 [9]. ICN models are perfect state plans of the Internet, where the object is situated at the central point of the scene. As opposed to the present Internet where packets are tended to as per source and destination nodes (i.e. IP addresses), ICN structures address content as per object name at the network layer. Each packet is named and this name serves to follow the objects. The point of the ICN proposal is to build up a design for the secure communication of content objects. ICN is one of the most significant researches for the design of future Internet[10]. ICN is a future Internet paradigm that focuses on information rather than location. In ICN, data is represented in the form of data objects or contents and all the contents have its specific global unique name by which users send their requests in for desired information and the network is responsible for the transmission of the requested information [11].

1.2 Introduction

Internet usage has increased extensively in the past decade, especially regarding the broad use of Video on Demand (VoD), which increases Internet traffic. The current Internet architecture supports end-to-end communication, and the content retrieval process occurs in two steps. First, the name resolution provides identifiers for the content (i.e., URL) with the location (i.e., IP address), and second, consumer Interests are transmitted based on the location from consumer to provider and Interested content from provider to consumer [12].

The name resolution occurs for the content at the application layer before forwarding the content which wastes network resources. For instance, if a copy of the required content resides at network router near the consumer, the consumer's Interest does not need to be transmitted to the main content provider, which decreases the usage of network resources and network traffic [13]. Moreover, the Internet is progressively recycled for information distribution with, relatively less use for end-to-end communication between end hosts [14]. Although Internet requirements have been increasing the evolution of Internet architecture has not followed the growing demands of Internet consumers and communications data [15]. For example, the existing communication worldview of the Internet presents numerous restrictions. First, the current Internet design offers numerous duplicates of objects, yet these duplicates are not connected. Keeping in mind the end goal to share the system's assets, extra overlay instruments have been proposed over the TCP/IP stack [16]. Reliable and fast object exchange requires application-specific systems, for example, Content Delivery Networks (CDNs) or Peer to Peer (P2P) [17] administrations. Second, security is accomplished via outsider applications and administrations. Establishing trust in receiving objects is difficult and the majority of associations depend on suspicious locations. Third, actually receiving objects is strongly related to the object's location. Each Internet packet follows the source and destination addresses. Every time an Interest packet is sent for object accessing, it needs addresses [18]. These Interests are firmly connected to an address, flagging a specific object location, even though consumers do not care about the location and just want the object to arrive as fast as possible. The direct manner to resolve this problem is to replace the "where" with the "what" [9]. Host-to-host correspondence was used to take care of the issues in the 1960s. Researchers contend that architecture using named objects is a better fit for the communication needs of today.

According to the Visual Networking Index (VNI) report, researchers have recognized other issues related to the IP network architecture that require quick comprehension [19] to have an acceptable platform that includes data integrity, confidentiality, availability, accountability (Owner/Publisher identification and authentication), and especially data transmission [20]. Several projects and proposals have been developed to improve Internet architecture, including ICN, which is a leading research focus for future Internet paradigm [9]. A number of ICN projects have been formed, such as the European Union Research and Innovation program (EU FP7). Under FP7 [21], several projects are being developed, for instance, Network of Information (NetInf) [22], 4WARD [23], Publish/Subscribe Internet Routing Paradigm (PSRIP) [20, 24], Publish/Subscribe Internet Technology (PURSUIT), Comet [19], COMBO, SAIL project, CONVERGENCE [24, 25], COAST [26], and Green ICN [27].

Similarly, the US research community has also initiated a number of ICN-based projects, including Content-Centric Networking (CCN), Named Data Networking (NDN), and Data-Oriented Network architecture (DONA). Another ICN-based project known as Active Content Management at Internet Scale (COMIT) was launched at University College London (UCL) in January of, 2014 [28]. All of these projects feature different functionalities and implementations. However, the basic purpose of all ICN-based projects is to design a network architecture that supports the named data distribution to handle the flash-crowd effect, disruption, and denial of service; reduce the use of resources and decrease energy consumption [11]. In addition, NDN provides the complete functionalities of ICN.

According to the latest forecast [29], global IP traffic will attain more than 1,000 ExaBytes (EB) per year in 2018. The majority of this increased traffic comes from peer-to-peer (P2P) IP infrastructures and a variety of different forms of video traffic, such as TV and VoD, which accounts for almost 90% of global customer IP traffic. The global mobile traffic is also estimated to increase at extreme rates at the same time. To deal with this enlargement in terms of data quantity and devices, one solution is to organize application-layer overlays, such as CDNs [30] and P2P applications, that cache content, offer location-free access to data, and enhance the data delivery method. This payoff should make access to named data or objects available, with replicated web resources, rather than the conventional host-to-host data delivery model. However, these procedures would reside in networks, and understanding the full potential of content-based sharing in today's IP-based platform is very complicated. NDN is an innovative networking approach to facilitate named data or objects as the supreme network entities [31]. It supports object naming and pervasive in-network caching to supply well-organized and vigorous network services [32].

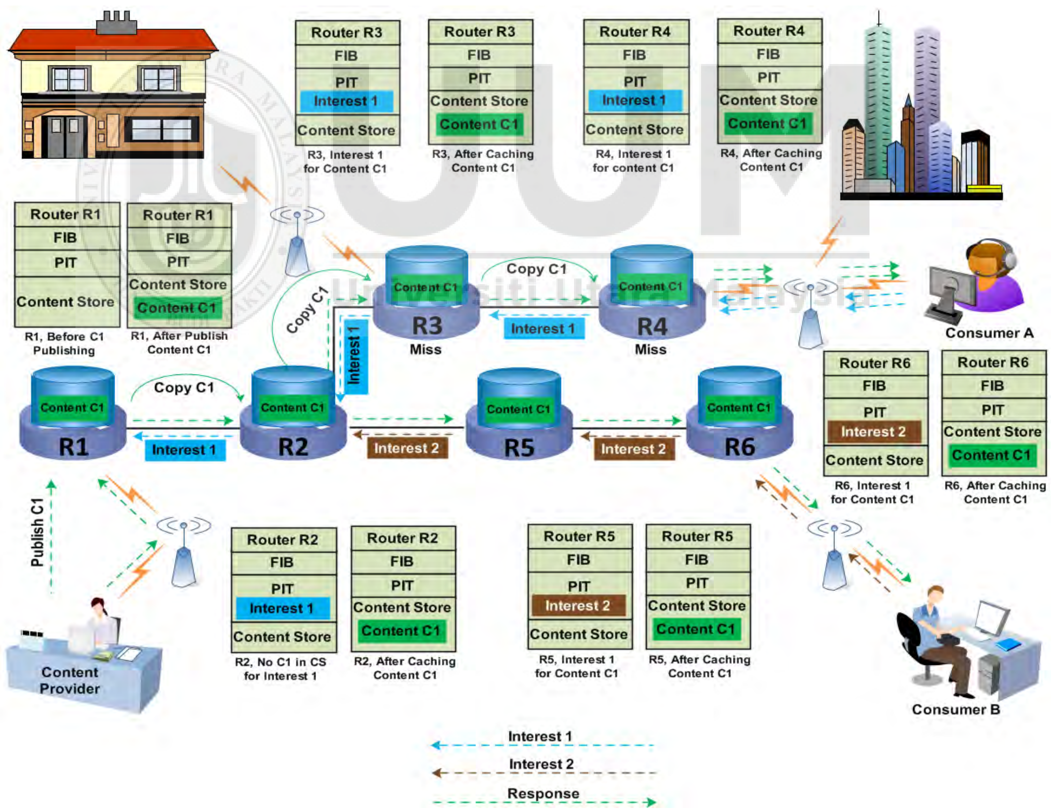

Figure 1.1. Named Data Networking Caching

NDN can be considered an enhanced generalization of CDN technology, and NDN operation is sufficient on a scale similar to CDN [33]. Moreover, NDN [34] is not limited to only media-sharing scenarios; it includes other schemes as well, such as data collection. For data dissemination, two types of message packets are used in NDN: Interest and Data (content). The Interest message with the name of the desired content transits the consumer's interest [35]. The Data, which retrieved from the content provider, is known as content in NDN. An Interest message identifies the Data packet (content) and retrieves it by specifying a full name with other restrictions that designate acceptable data. A Data packet (content) message holds the required information payload and the identification of the provider.

In Figure 1.1 the NDN content caching mechanism is demonstrated as Consumer A sends out Interest1 for Content C1. C1 is found at router R1 because the content provider already published C1 at R1; therefore, R1 now becomes the intermediate provider. Consequently, R1 instantly responds to Interest1 by sending the desired Content C1, and a copy of C1 is cached locally during its transmission at all on-path routers (R2, R3, and R4). A subsequent Interest2 is generated from Consumer B to retrieve C1. As a result, C1 is found at R2 at this time, and R2 sends C1 to Consumer B with multiple caching operations along the data routing path (R5 and R6), as illustrated in Figure 1.1.

To summarize, the NDN project has been developed by combining several modules in which caching is considered the most dominant component for actualizing the NDN manageability [36]. The Cache is used to store the transmitted content in two different forms: on-path and off-path [37]. These caching techniques can be implemented efficiently with the help of content deployment strategies. In addition, a basic need behind caching performance is determining the appropriate position to place the transmitted content. Consequently, many cache deployment strategies have been proposed for achieving some advantages, such as proper use of bandwidth by enhancing the appropriate delivery of information and reduction of delays [38], as well as minimization of the overall load on the main server (content provider).

1.3 Motivation

The present IP-based Internet was originally developed as a part of military experiments considering that the target shares its resources owing to the scarcity and expensive cost of resources. Therefore, considerably fewer people had access to it [39]. Now, the circumstances have changed; the cost of Internet devices has dropped; further, the number of Internet users has exceeded 4.02 billion. Consequently, currently, one device per capita is measured and this number is predicted to increase to three devices in 2010.

However, the number of users is not the only aspect that has brought change to the Internet. The availability and transfer of data content through the Internet have been increasing rapidly for many years. Google has already indexed one trillion web pages [40]. In every 5 min, 3.9 PB of data is being transmitted across the Internet. In 2018, smartphones transferred 9 EB of data monthly and these numbers are expected to keep increasing. In 2019, networks will deliver 14.5 PB in each interval of 5 min and smart-phones will transfer 17 EB per month. The VoD traffic is recorded at approximately 4.3 billion and it is expected to reach 7.2 billion DVDs per month by

2021. Consequently, the current Internet architecture cannot fulfill the requirements to handle these exponentially growing Internet users and data transfers.

In short, the Internet has evolved from serving a few selected groups of scientists to reach more than 3 billion users [39]. The Internet has progressed from serving nonprofit scientific and military activities to becoming an important part of the world economy. Indeed, it has been recorded that the Internet has contributed to 4.5% of the GDP in both G8 and BRICS countries [40]. It has adapted from the exchange of static files and messages to recording hours of digital videos and multi-party video conferences. Nevertheless, the communication architecture has remained unchanged and the network communication procedure is still directed by a peer-to-peer model. Therefore, the current communication design of the Internet has been encountering several limitations regarding the physical location based on data transmission. For instance, the Internet architecture can deliver multiple copies of similar data items owing to the nature of its communication architecture. To share resources of the network, supplementary overlay mechanisms were proposed over the TCP/IP stack. Fast and reliable data transfer requires an application-specific mechanism in which content delivery networks [41] or peer-to-peer services are primarily used. Another serious concern is a security protocol that has been achieved by using third-party applications and services. It is not easy to trust the retrieved data item and most of the connections rely on untrustworthy locations [42].

As illustrated in Figure 1.2, the diverse Internet users exhibit an inclination for multimedia data and shared videos. For example, YouTube, which is the 3rd most visited website around the globe, has recorded that approximately 5 billion videos are watched in a single day and approximately 6 billion hours of videos are viewed in a month; further, approximately 400 hours and 700,000 hours of videos are uploaded and watched per minute, respectively [43]. Moreover, 10 million videos are viewed using mobile phones and the annual management and maintenance cost is approximately $63 billion. Similarly, the video provider NETFLIX has approximately 86.7 million subscribers; moreover, 10 billion hours of videos are viewed per month or 100 million hours of videos are watched in a day. In addition, more than 100 hours of videos are uploaded, and 78,000 hours of videos are viewed on NETFLIX per minute. The total videos watched on YouTube and NETFLIX will approximately be equal to half the traffic on the Internet by 2021. In 2020, five years would be required to watch all the videos that will be transmitted on the IP network per second [44].

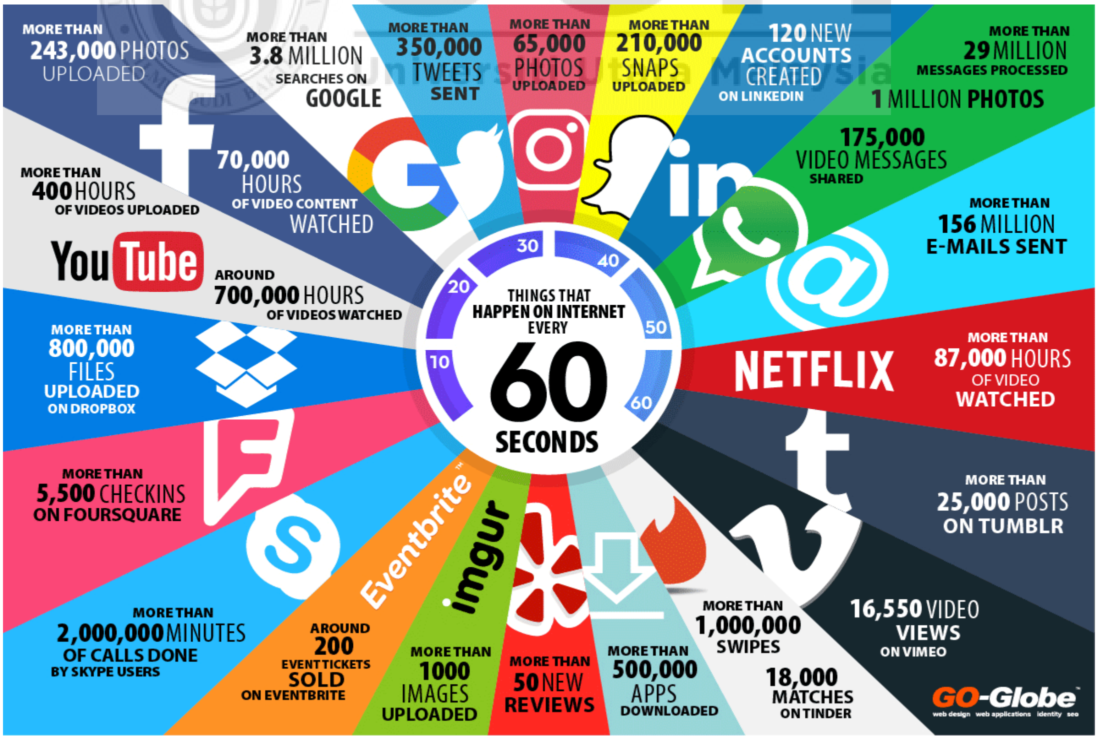

Figure 1.2. Internet in a minute [45]

Today, people are significantly interested in social media and have several social relationships where information is sent and received using social sites. For example, Facebook is one of the most popular social sites with approximately 1.71 billion global users that generate four new petabytes of data per day. Furthermore, approximately 100 h of videos are watched and more than 350 billion multimedia contents are uploaded in a day [46].

The Internet has adapted from trading of static documents and messages to serving hours of advanced video and multiparty video meetings. Despite all these changes, the correspondence worldview has remained unaltered and the system operation is still driven by an end-to-end model. In addition, its integration has been broadly acceptable to produce new technologies and directions like cloud computing, distributed fog computing [47], ubiquitous computing [48, 49], and fly computing [49]. However, it is crucial to continue the efficient usage of the Internet because the global traffic is expected to increase exponentially in a few years [50]. According to the latest forecast on the sharing of information and data traffic on the Internet per minute [51], approximately 35.7 million messages are transmitted currently on the Internet per minute. Google has indexed more than 1 trillion web pages [52].

Moreover, in every minute 1.2 PB of traffic is communicated through the Internet and 156 million E-mails are sent. Furthermore, Google is used as the primary search engine, which demonstrates a high rate of hits of approximately 4.3 million searches performed per minute. According to the recent report of the Cisco Visual Networking Index (VNI), the demand for information per minute is considerably high. At present, users prefer multimedia-driven content rather than ordinary file transfers. Further, more than 39,300 hours of music is listened to with 14 new songs being added on Spotify per minute. Moreover, more than 1 million photos, 175,000 video messages, and more than 29 million instant messages are shared on WhatsApp every minute [53]. In addition, around 56,000 photos are uploaded on Instagram per minute.

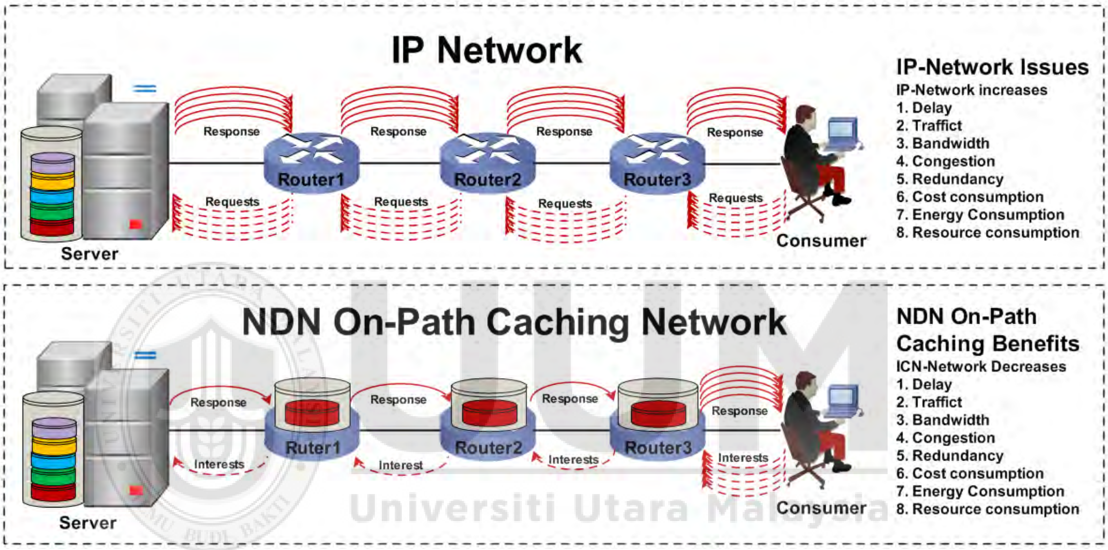

Figure 1.3. IP-based Internet versus Named Data Networking architecture

Thus, for the current Internet architecture, such a large amount of multimedia data is significantly difficult to handle. This is because the retrieval of a data item is strongly associated with its physical location. The packets of the entire Internet are required for using the source and destination addresses and each data item is retrieved using its location address. In IP-based Internet, the requests of the consumer are strongly associated with an address representing a location, whereas for caching in Named Data Networking (NDN), consumers do not focus on the location but only on obtaining the desired data items as fast as possible. Subsequently, the NDN approach strongly criticizes the use of physical locations because consumers are interested only in the desired data rather than locations. Hence, in this scenario, the location of the data is replaced by the information regarding the data that is required by the consumers; thus, this concept overcomes the limitation of the host-to-host communication and solves the problems of the 1960s [1].

Figure 1.3 illustrates the IP-based Internet architecture and trending problems, which are expected to be uncontrollable in the future. The nature of the current Internet is to send similar objects over the Internet multiple times that result in an increase in the delay in response because every time the user sends a request, the request has to traverse to the server. Therefore, the path length between the user and server increases causing a delay and decrease in the resource usability. Moreover, as Internet channels have lower capacity when compared to the amount of data transmitted, the congestion is uncontrollable. In addition, multiple similar requests generated by different users increase the redundancy in requests. Consequently, all the requests consume extra energy (power consumption). The Internet of the near future demands a conversion from the host-centric to an information-centric paradigm [54]. The significant problem for the current Internet is that one server is required to send the same data multiple times, which causes 10% of the transmitted data resulting in 90% of the network traffic. This massive amount of traffic can be managed by implementing cacheable routers intelligently in the network using the concept of on-path caching [55].

Consequently, the on-path caching mitigates the constant overloading owing to increasing traffic on the network and provides better performance than the IP-based Internet. A part of the rationale is that when a query is served, the cache of the router is enabled to store the delivering content for further communication in lesser time with lesser redundancy along the path, which reduces the traffic load [56]. The arising problems discussed previously have motivated this research to adapt to certain Internet paradigms that can solve these problems with high performance and will be efficient for future requirements. Moreover, several projects were developed to improve the Internet architecture in which on-path caching was highly recommended by the researchers for the future Internet paradigm.

Another rationale for implementing this method is that the rapid growth of network information on the Internet required approximately 500 EB in 2010 and it is expected to reach 15.3 ZB by 2020 [57]. Thus, the IP-based Internet traffic is exponentially increasing, which will cause significant blockages that will be a serious problem for the future IP-based Internet architecture [20]. Furthermore, there is a requirement to identify an alternative platform for the present information dissemination mechanism. The Information-Centric Networking Research Group (ICNRG) [9] concluded that currently the distribution and manipulation of information is a responsibility of the Internet rather than the data source [58]. Consequently, it is an essential requirement of the future Internet architecture to minimize the present challenges. The objective of ICNRG is to develop a scalable network system by the implementation of an on-path caching system within the network.

1.4 On-Path Caching Design Concept

On-path caching is the basic approach that is used to cache content within the network for a specific time span. In this approach, the node responds to the consumers using locally cached content. The node transmits the corresponding content to the consumer when it receives an Interest. All the NDN nodes have the ability to perform a matching operation between the received Interest and the cached contents [59]. In on-path caching, data communication is performed considering two primitives such as Interest and content. The Interest carries the name of the required content in the prefix. A consumer broadcasts the Interest to the network and the Interest is forwarded by using the Forwarding Information Base (FIB) record to identify a suitable content container. A node having the required content can respond directly to the consumer by transmitting the corresponding content. The content is transmitted as a data packet.

In NDN-caching, consumers can send their Interests for a particular content through multicast, broadcast, and anycast data transmission modes. The content is transmitted only when an Interest packet matches with the data packet [19, 60]. The caching system of the NDN router has three components: content store (CS), pending interest table (PIT), and FIB. The CS represents the cache used by NDN routers to store a copy of the disseminated contents during its transmission between the provider and the consumer. As a router receives an Interest for specific content, the CS compares the received Interest with the existing entries of cached contents. If any of the existing entries match the received Interest, the CS transmits the corresponding content immediately back to the consumer by following the reverse path of the path followed by the received Interest. Caching a copy of the transmitted content occurs within the router to serve subsequent Interests. The PIT is responsible for managing all the Interests received from the users and gathering them into a table, if an entry for the received Interest does not exist, to make the back routing easy [61, 62].

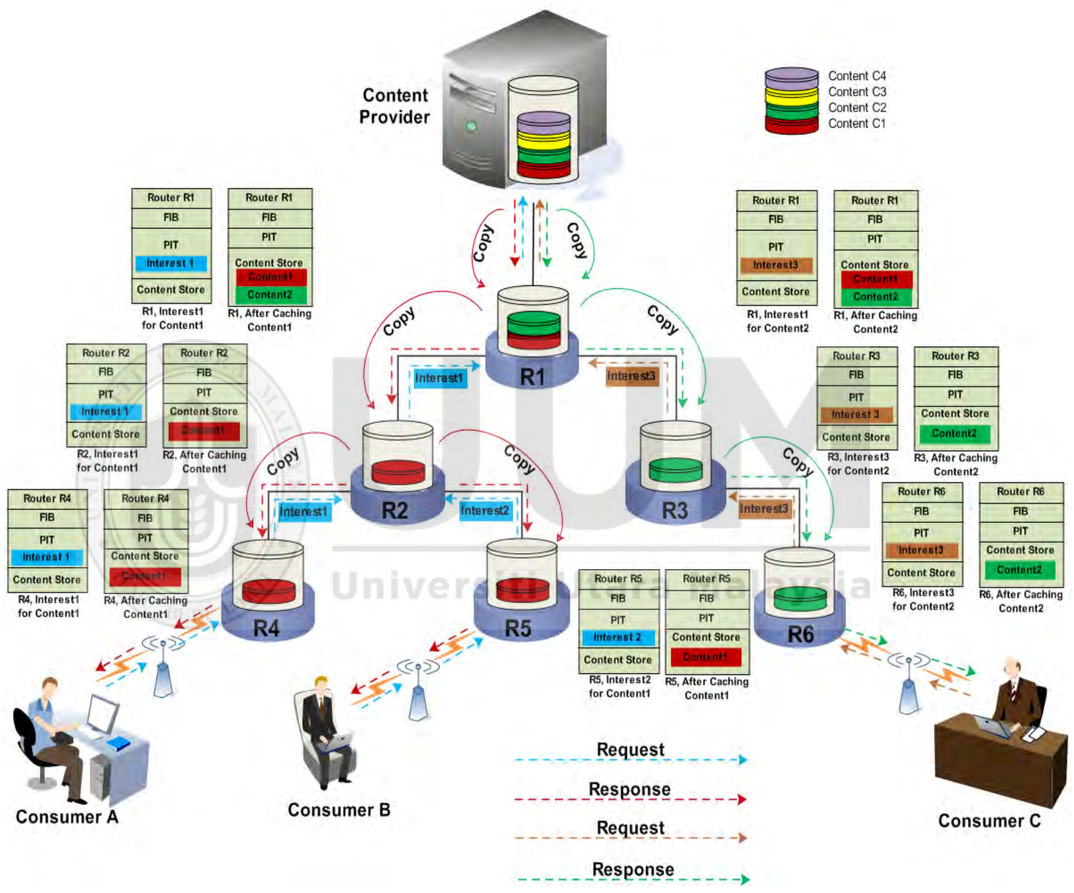

Figure 1.4. Named data networking on-path caching mechanism

The PIT keeps track of ongoing Interests so that the data can be sent back to the appropriate consumers. If the received Interest fulfills its requirements from the locally cached content, the corresponding content is sent back to the consumer through the same path. Conversely, if the Interest packet does not match with any of the existing entries, the Interest is forwarded towards the suitable content container. FIB contains the routing information that is used to construct a routing map of the network topology. It provides a suitable direction for the Interest packet to be forwarded in a network. It also helps the Interest packet to identify a suitable interface to forward the Interest of the user towards the appropriate content container [63].

An example of the design of an on-path caching architecture and its data dissemination is illustrated in Figure 1.4. In the given example, consumer A sends an Interest1 to retrieve the desired content C1. When Interest1 reaches router R1, and the required content (content C1) is obtained, then the router R1 becomes a provider and transmits the desired content C1 to consumer A. However, if the requested content is not obtained at R1, the router forwards the Interest towards a suitable source (content provider P) and an entry will be created in the PIT. The PIT entry will be deleted when it is identified that the required content is cached in its own CS. For the first Interest of consumer A, the request traverses to router R1, where it is satisfied by obtaining the required content C1. Then, a copy of C1 will be cached at the intermediate router R2 during its transmission from R1 to consumer A to reduce the distance between the provider and the consumer for subsequent Interests received at R2 from consumer B. Consequently, the consumers located around consumer B will receive the content C1 from router R2 and not from router R1.

1.5 On-Path Cache Management Strategies

Cache management is very important because on-path caching cannot generate efficient results. Cache-less networks can produce efficient results. The management of the cache depends on how the cache is deployed and managed by the network. The effective management of the cache demonstrates a mechanism that places a copy of the transmitted content into the network caching system efficiently [14]. This mechanism reduces the server load and latency of subsequent requests. Further, it increases the cache hit rate and availability of data [15]. Therefore, the intention of on-path caching is to reduce the usage of the resources of the Internet. For this purpose, on-path caching delivers several cache management strategies that help in overcoming the data transmission problems such as bandwidth consumption, content redundancy, and path length between the publisher and the consumer.

For efficient caching distribution, several placement strategies were proposed. To understand the idea of on-path caching, it is important to explain the mechanisms that are required to functionalize the cache for content caching. Thus, a number of diverse types of cache management strategies are recently been designed such as; MAx-Gain In-network Caching (MAGIC), WAVE Popularity-based Caching Strategy, Hop-based Probabilistic Caching (HPC), LeafPopDown, Most Popular Cache (MPC), and Cache Capacity Aware Caching (CCAC).

These caching strategies enhance overall data dissemination services. However, these strategies still facing some critical problems such as; how each node makes a caching decision is a major problem in on-path caching that impacts the performance of the content delivery process. For example, popular content needs to be cached at the node where it will be demanded next. The information content can be cached opportunistically at the delivery path for the consequent Interests [64]. The efficiency of caching is dependent on different types of performance metrics that are necessary to improve the content caching strategy by considering the cache size, which is small compared to the communicating data content. Therefore, some essential performance metrics such as cache hit ratio, content diversity, stretch, and content redundancy are included in this study to critically compare types of caching strategies to determine the optimal caching.

1.6 Problem Statement

NDN caching brings a revolution in network architecture. It can overcome the location-based issues that arise in the current Internet architecture. However, it is difficult to decide which content needed to be cached and at what location to improve the network performance. Cache-management strategies have been developed to achieve efficient results. Still, it is difficult to decide which caching strategy is the most ideal for each network environment.

All the popularity-based caching strategies focus on the most popular contents and only the contents are selected as popular that has received the maximum number of Interests. On the other hand, some popularity-based caching strategies such as MPC and LeafPopDown have a very small threshold that caused the increasing of caching operation by less popular contents because it assigns the same priorities to all selected contents. However, HPC, WAVA, CCAC, and MAGIC select the content as popular that has received maximum number of Interests. Therefore, if a content is located far from the desired consumer, and numerous Interests are received for that content, but the popularity of that content is lower as compared to the popular content, the content will never suggests as a popular content and all the times, the incoming Interests for that content needs to traverse the several hops to find the required content from remote providers. The reason is that, the content selection mechanisms do not allow the content having slightly lower popularity than the threshold to be cached. For instance, the threshold is set as 1000 to select a popular content than all those contents which have lower popularities will never supposed to be cached at intermediate nodes. Therefore, all these strategies focus on the most popular or most recently interested content that cannot fulfill the efficient threshold requirements for the most popular and less popular contents. Hence, the content diversity is reduced because numerous redundant caching operations will be occurred by the similar contents [65, 66].

Moreover, HPC, CCAC, WAVE, and LeafPopDown [67] increases the redundant content replications at all on-path routers. [19]. These strategies need additional parameters such as TSI, TSB, CCV_i, CCV_{th}, additional FIB, w_r to be computed for all incoming Interests and contents at each router. All these entities execute at all the routers separately whenever an Interest is generated, or a content is transmitted to the consumers. In this way, the communication overhead is increased and consequently, the cache hit ratio keeps at its minimum level because millions of Interests are generated and correspondingly contents are transmitted in a very short interval. Moreover, similar types of popular contents are cached at multiple nodes that decrease the overall diversity ratio. MAGIC uses extra cost and resources to compute max-gain and local gain values. Consequently, it reduces the cache hit ratio because it caches the popular content at limited locations. Hence, most of the incoming Interests need to forward to the remote providers. The problem statement is summarized as follows; all these strategies focus on the popular or most recently interested contents that cannot fulfill the efficient threshold and content caching requirements for the most popular and least popular contents that caused multiple homogeneous contents' replications along the publisher-subscriber path. Consequently, the overall caching performance is reduced in terms of caching hit ratio, stretch ratio, content redundancy, and diversity ratio. However the proposed caching strategy not only prefers the most frequently interested content, but also handles the less popular contents to be cached at an appropriate location to increase the overall caching performance.

1.7 Research Questions

The purpose of this research is to design a new cache management strategy to enhance the content caching in terms of cache consumption of a network node. Regarding the critical issues with the existing caching strategies as mentioned in the problem statement, the following questions are needed to be identified:

1. How the threshold regarding Interests frequency for content would be improved to aid better caching system?
2. How would the selection of content caching position be improved for cache effectiveness in NDN network?
3. What will be the consequences of proposed cache deployment strategy on caching performance in NDN-based network environment?

1.8 Research Objectives

The main objective of this study is to design a cache management strategy for NDN, named as Compound Popular Content Caching Strategy (CPCCS). This strategy is proposed to enhance threshold and the intermediate position for content caching to improve the performance in terms of diversity, redundancy, cache hit ratio, and stretch ratio.

The intentions of CPCCS can be specified by the following objectives as given below:

1. To design a new content selection mechanism that will obtain a relationship between content popularity and Interests' frequency using an algorithm.
    a) Differentiate the contents regarding their Interest frequencies.
    b) Select the contents as optimal popular and least popular using threshold.
2. To develop a position selection mechanism in order to optimize the content store of a network node.
    a) Propose relationship to get the content caching positions for optimal popular contents between publisher and subscriber.
    b) Find a cache positions in order to get efficient usage of a network node by caching least popular content at publisher-subscriber path.
3. To validate and evaluate the performance of the proposed caching strategy in NDN network simulation environment.

1.9 Research Scope

The purpose of this study is to design a content placement strategy named as Compound Popular Content Caching Strategy (CPCCS) for the management of NDN cache. The basic concept of proposed strategy is to focus on the selection of contents regarding their recent utility to increase the amount of diverse content with minimum number of redundant content replications within network nodes and find an appropriate position for caching of content to enhance the cache management in order to achieve better cache hit ratio with short stretch path. Diversity, redundancy, cache hit ratio, and stretch are the common metrics used to measure the caching performance in NDN.

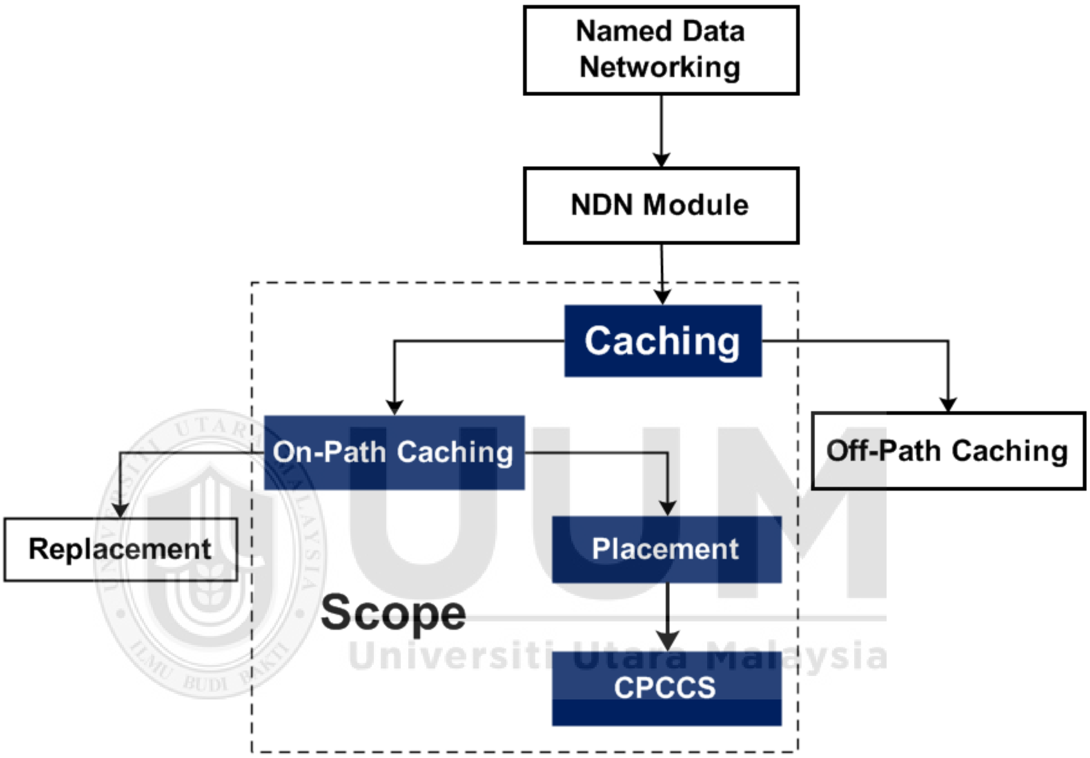

Figure 1.5. Scope of the study

Several studies recently been published related to NDN caching strategies [46, 60, 68], provide information regarding the caching decisions for disseminated data content [56, 69, 70]. However, the present research is conducted to determine caching performance based on the most relevant basic metrics for improving the overall caching performance. Moreover, this study focuses on the critical analysis of popularity-based caching strategies and the impact of basic caching evaluation metrics, including content redundancy, content diversity, Stretch, and most importantly, cache hit ratio. In addition, the proposed strategy is extensively and comparatively studied with other NDN-based caching strategies such as HPC [71], MPC [72], LeafPopDown [68], MAGIC [73], CCAC [74], and WAVE [75] through simulations.

1.10 Significance of the Study

The CPCCS will be useful to control the network congestion by caching the disseminated data items at suitable intermediate locations. Therefore, it can deliver several benefits such as low response latency, low bandwidth, low congestion, and low power and resource consumption. Consequently, it achieves easy control of data traffic. Moreover, the intermediate location for the cache is beneficial for mitigating the use of resources and energy consumption and the total transmission cost can be maintained at its minimum level. Consequently, the proposed caching strategy mitigates the constant overloading owing to increasing traffic on the network and provides better performance than the IP-based Internet.

A part of the rationale is that when a query is served, the cache of the router is enabled to store the delivering content for further communication in lesser time with lesser redundancy along the path, which reduces the traffic load. In other words, if a content resides on the main server and it is very popular, for example, the server receives a lot of requests for that content. Then, each time the server needs to send that requested content to the users that will cause maximum bandwidth utilization. On the contrary, if the contents are cached along the path, then all the coming requests will be satisfied by the locally cached contents and maximum bandwidth can save as well as content retrieval time will be minimized.

1.11 Thesis Organization

This thesis is organized in the following seven chapters according to the highlights given below:

Chapter One It initialized with the background of the Internet and introduces NDN with a brief explanation of NDN characteristics. Moreover, the motivation for this study is comprehensively explored. Likewise, it also contains a problem statement, objectives, scope, research steps, significance, and key contributions of the study.

Chapter Two reviews the past literature related to the study. It also explains the key terms with advantages and drawbacks that still exist in the field of Named Data Networking cache deployment mechanisms.

Chapter Three refers to the sequence of steps to follow the particular methodology that is used in achieving the objectives. It also outlines the tools that are considered for simulation purposes.

Chapter Four illuminates the design of content selection mechanism Compound Popular Content Selection (CPCS) to accomplish the first objective. Likewise, it also describes the proposed content caching mechanism named as Compound Popular Content Caching (CPCC) that is designed for cache manageability along the data routing path. As a result, both proposed content selection and content caching mechanisms are extensively and comparatively studied with other NDN-based caching strategies to evaluate the performance in terms of diversity, cache hit ratio, stretch, and redundancy.

Chapter Five provides the performance evaluation through taken by the comparison of the proposed strategy with others using different parameters. The simulation environment is selected to check the effectiveness of the research via a diverse perspective and network topologies.

Chapter Six concludes the study by giving a summary of the research, contributions, and explore suggestions for future research direction.

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

The existing Internet was not designed for the applications that were recently developed such as voice over IP and network address translation. To solve these problems, it is crucial to modify the Internet architecture [76] as several presents and unresolved problems (bandwidth and power consumption, congestion, security) still do not have satisfactory solutions in the existing Internet. These concerns necessitate the implementation of a new technological approach to achieve an efficient Internet architecture that can satisfy its current requirements [77]. For this purpose, the research communities have proposed several architectures, of which the most popular is the named data networking (NDN) architecture owing to its versatility [78].

NDN is highly scalable, efficient, and reliable information distribution network architecture. These advantages have encouraged most researchers to modify the current end-to-end sender driven Internet paradigm to a receiver-driven information-centric paradigm. The NDN architectures provide on-path caching (storage for data objects) in the entire network and multicast communication can be produced through data replication.

The objective of on-path caching in the NDN architecture is to achieve a scalable, effective, and consistent distribution of information and data objects by using a common communication platform that is available in a dedicated system like a content distribution network [20]. In fact, there is a requirement to modify the present working IP-based network architecture to the NDN architecture [79]. The on-path caching techniques in NDN [80] provide the advantages of disseminating information [81][82]. As the Internet is progressively improving in term of information distribution, there is a relatively lesser requirement for end-to-end communication between end hosts.

The NDN architecture is a receiver-driven networking approach where end-users express their interest for specific content and the network is responsible for routing the user names-based requests towards the appropriate content container [83]. The NDN architecture implements on-path caching (generally known as storing) to facilitate multicast mechanisms. Thus, it deploys efficient and appropriate transmission of the desired information to the end-users. In this situation, temporary storage servers are generally utilized all over the Internet to serve the requested objects [84].

In the current Internet, the number of data items is significantly large when compared to the number of IP addresses, which implies that it is challenging for the IP address-based architecture to disseminate the desired data in time [85, 86]. The on-path caching in NDN can resolve the problems arising in the IP-based Internet by implementing the cache storage within the network nodes. This technique is useful to control network congestion by caching the disseminated data items at intermediate locations [87]. Therefore, low network congestion needs lower cost and reduces response latency. Consequently, it achieves easy control of data traffic. Moreover, the intermediate location of caching is beneficial for data transmission because it mitigates the use of resources and energy consumption [88, 89].

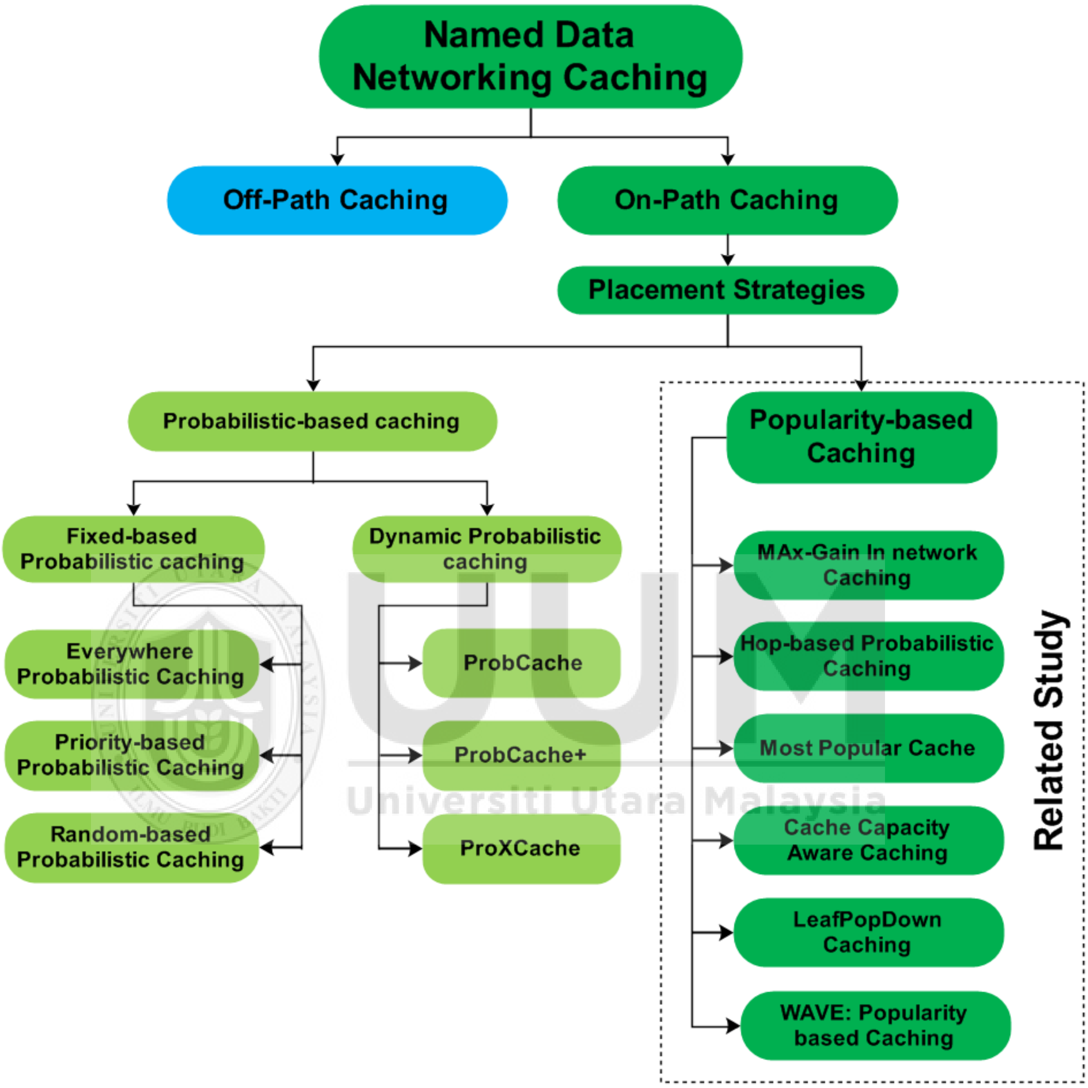

Figure 2.1. Classification of literature and related study

2.2 Named Data Networking Caching

NDN is an innovative networking approach to facilitate named data or objects as the supreme network entities [31]. It supports object naming and pervasive in-network caching to supply well-organized and vigorous network services [32]. Moreover, NDN [34] is not limited to only media-sharing scenarios; it includes other schemes as well, such as data collection. For data dissemination, two types of message packets are used in NDN: Interest and Data (content). The Interest message with the name of the desired content transits the consumer's interest [35]. The Data, which retrieved from the content provider, is known as content. An Interest message identifies the Data packet (content) and retrieves it by specifying a full name with other restrictions that designate acceptable data.

A Data packet (content) message holds the required information payload and the identification of the provider. NDN router consisting of three types of data structures such as; Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB) which is used to facilitate ubiquitous caching and loop-free forwarding. CS indicates the cache storage that holds the content for subsequent retrieval. It is managed via cache placement/replacement algorithms [55] that exploit the possibility of data reuse. The PIT is used to record unsatisfied (pending) interests. Each entry in the PIT has a list of source entries incorporated with incoming faces to the NDN router/node. The PIT has a specific timeout interval for unsuccessful Interests to avoid maintaining previous Interest records. The FIB is used to sustain the topology structure in tabular form to facilitate outbound face information for Interests. These tables are used for longest-match lookups in the prefix order through the content name. Each FIB entry presents a set of corresponding faces rather than a single face. In NDN, Interest packets are forwarded all the way through FIB, but the Data packets are forwarded according to the PIT records, which direct Data packets back to consumers. Consequently, an outstanding

FIB design is necessary for the transmission of both Interest and Data packets in NDN [87].

Caching is used to diminish the average cost (energy and time) and enhance the content retrieval procedure because a cache stores a copy of frequently interested data temporarily for subsequent Interests. A prominent example of this storage is web caching, in which web documents (e.g., HTML pages and images) are stored to mitigate issues in bandwidth use, server load, and reply time. Caching makes NDN different from the currently IP-based Internet, where the cache is used to transmit content to the consumers next NDN cache minimizes the response time, lessens resource consumption, and boosts data communication services [88, 89]. Suppose, a consumer sends an Interest for a specific data content to its nearest router with the corresponding content in its cache [90]. The intermediate routers become provider whenever disseminated content passes through their cache, and they store the passing content locally in their cache for subsequent Interests. Therefore, the provider router sends the required content to its desired consumer. The caching is divided into various categories based on their characteristics. It is hard to find an appropriate criterion that will best meet all future requirements. Basically, NDN caching is divided into Off-path caching and on-path caching, as described next as shown in Figure 2.1.

2.2.1 Off-Path Caching

Off-path caching is similar to the traditional proxy caching or CDN server. The retrieval of the content in off-path caching requires redirection of Interests. For off-path [89] caching a special entity, called the name resolution system (NRS), must be integrated. All sets of content are registered in NRS at the time of creation, and consumers can send their Interests for registered content [91].

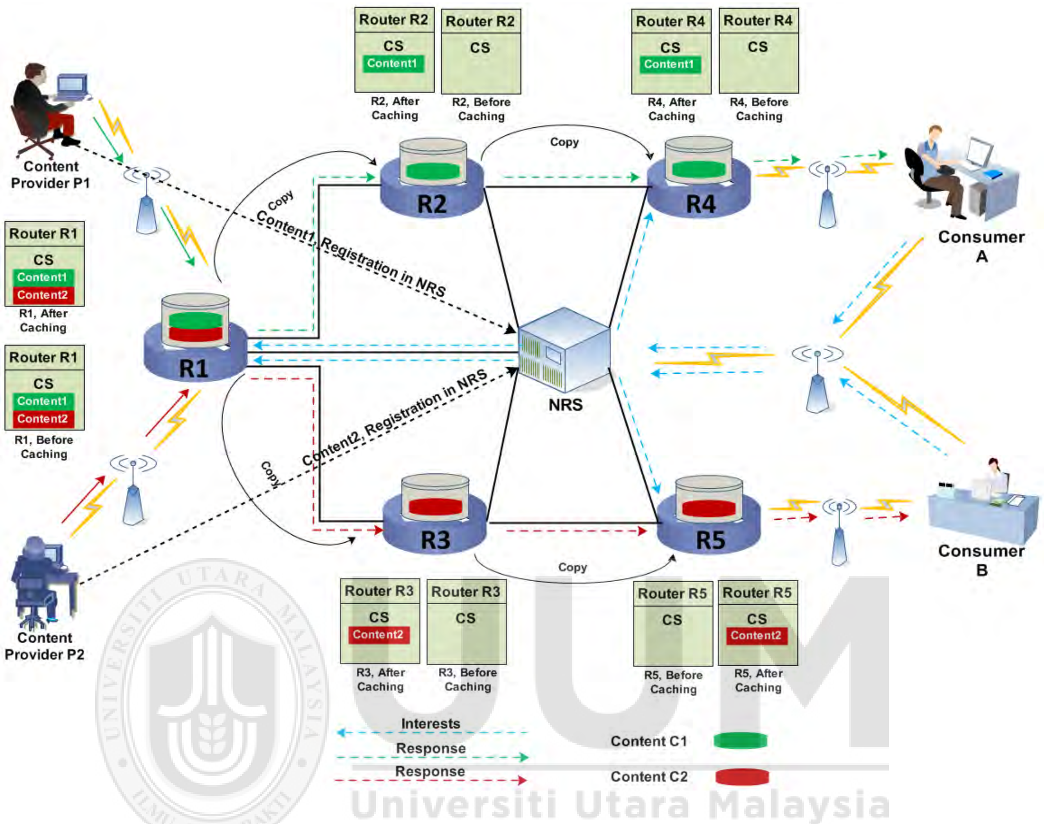

Figure 2.2. Off-path caching

The general problem with off-path caching is choosing what to cache and where to cache it. The other issue is how to reduce the overhead to inform the NRS when a new item is cached or an old item is discarded. The exact information about the content depends on the NRS, which needs to be updating continuously with all the information related to the registered content. In off-path caching, a consumer sends an Interest to the NRS, and the NRS compares it with the available content information [90]. Afterward, the Interest is forwarded to the corresponding provider (router), and the provider sends the required content to the consumer following the returning path, as shown in Figure 2.2. As the content is registered in NRS, the consumers send their Interests to the NRS. After that, the NRS forwards the received Interests to the appropriate providers (e.g., router or main content provider P1 and P2). The subsequent content is sent to the desired consumer through NRS or directly to the consumers. Consumers A and B send their Interest packets to the NRS to retrieve Content1 and Content2. The NRS redirects the received Interests to the appropriate content provider (e.g., router R1). R1 responds by sending the interested Content1 and Content2 to the appropriate Consumers A and B. The transmitted sets of content are cached along the data routing path during their dissemination from providers to the consumers; for example, Content1 is cached at routers R2 and R4, and Content2 is cached at R3 and R5, respectively, as illustrated in Figure 2.2.

2.2.2 On-Path Caching

In the on-path approach, when a caching node entertains a consumer's Interest, it responds to the interested consumer with a locally cached copy without NRS participation [92]. Therefore, this approach diminishes the computation and communication overhead during insertion of transmitted data in the network [19, 93]. When the Interest is projected in-network, the dominant provider (router) instantly responds to the received Interest by sending a copy of locally cached content to the appropriate consumers. Figure 2.3 illustrates the caching scenarios as follows: Consumer A sends out an Interest to retrieve Content1, which is found at router R1. Consequently, provider router R1 instantly responds to Consumer A by sending interested Content1, and a copy of the demanded content is cached at all intermediate routers (R2 and R4) during its transmission from provider router R1 to Consumer A.

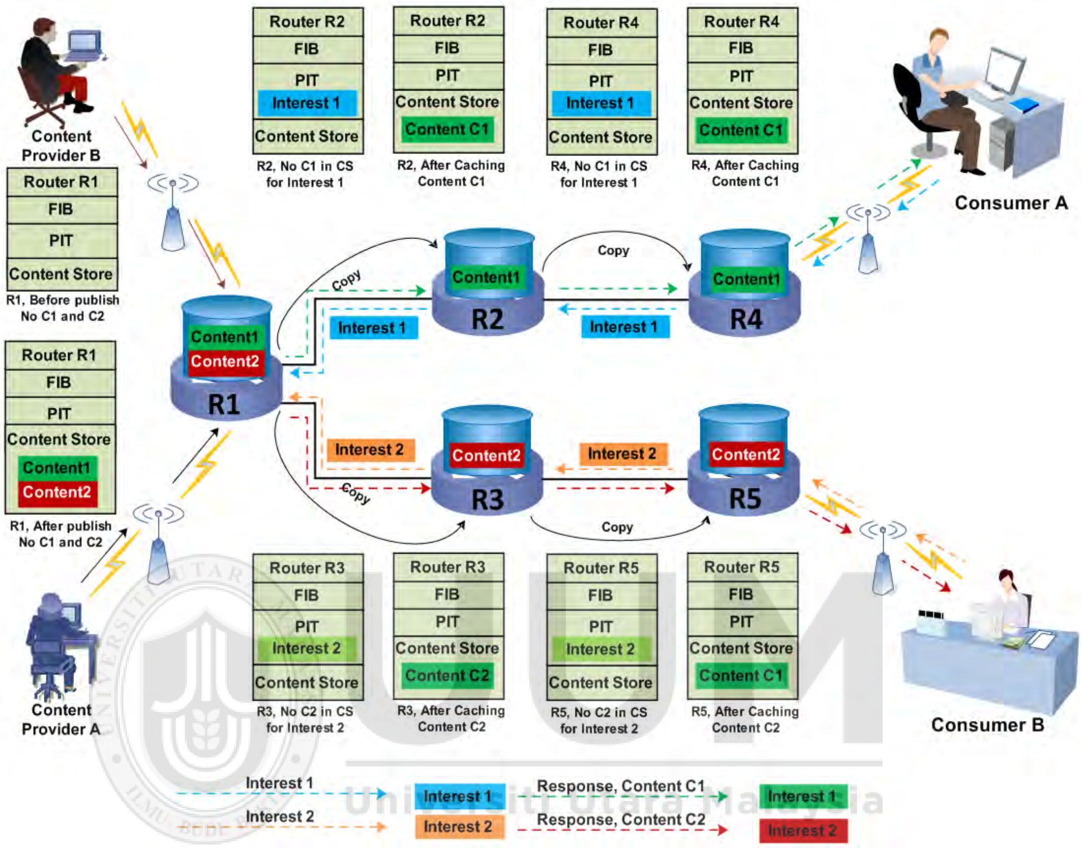

Figure 2.3. On-Path Caching

Caching mechanisms are divided into various categories, according to their algorithms [94]. Among them, on-path caching is the basic approach of NDN architecture, in which disseminated content is cached along with the data downloading path as it travels from the source to the destination [95, 96]. Therefore, this approach is useful to reduce the computation and communication overhead during the caching of disseminated content along the delivery path. A significant issue related to on-path caching is how each node makes the caching decision to improve the content delivery process. For example, popular content needs to be placed at the node from which it is next going to be requested [90, 97].

The caching decision is the most important part of communication in NDN because all caching abilities will suffer if the caching decision is wrong [69, 98]. Although information can be opportunistically cached on the data routing path, subsequent Interests for the same content will follow the name resolution path and reduce the possibility of a cache hit [97, 99, 100]. When data routing and name resolution is coupled, and data is cached on the data routing path, as cache hit occurs. On-path Caching has a significant role in increasing overall network performance, i.e., reducing bandwidth consumption and minimizing latency during access to the desired content [55, 101]. It also reduces server load and maximizes the availability of the desired contents [102]. The objective of the content placement approach is to decrease the distance between the consumer and the content provider. If the content resides at an intermediate node the stretch and content retrieval latency will be kept at a minimum level [103, 104].

On-path caching is a compatible module of NDN handled by using cache management strategies [105]. Caching of content is based on different criteria, such as caching decisions based on node centrality, distance from the source, and the content's popularity [106, 107]. According to the naming granularity, caching can be categorized into several types, for example, packet-level caching, chunk-level caching and object-level caching. According to the nature of the caching model, the strategies are categorized as autonomous model-based caching strategies, cooperative-model-based caching strategies, and centralized model-based caching strategies [75, 108, 109]. These strategies have reduced the usage of network resources, and minimize the network traffic. However, it is very difficult to provide such a large amount of cache storage [110, 111]. On the other hand, efficient content dissemination can be provided by selecting a suitable caching strategy that can distribute contents with minimum usage of resources (caches) [112, 113]. In the default NDN caching strategy, when a consumer sends an Interest to the network, the network sends the required content to the consumer and a copy of that content is cached along with the data delivery nodes available on the publish-subscribe path. Thus, subsequent Interests should be satisfied locally.

2.3 NDN Cache Deployment Strategy

Cache deployment deposits the consumer's desired content along the data routing path in provisional storage (cache) that is allocated within the network nodes. Caching plays an immense role in network performance by diminishing bandwidth consumption and reducing latency when accessing the required content [114]. An essential operation of these caching strategies is forwarding the desired content to the interested consumers and caching a copy of the content near the interested consumer to minimize the average cost for subsequent Interests (requests). Recently, a number of content deployment strategies have been projected to deploy NDN in-network caching. These strategies boost information replication and decrease reaction time [115, 116]. In addition, strategies use fewer network resources and reduce network traffic [117]. For example, probabilistic caching [118] is a well-known content caching strategy, and the research community captures more interest because of its adjustable probabilistic value [119], which is allocated according to the manufacturer. In recent years, probabilistic caching got incredible interest from the research community because of its diverse natures of caching content. Generally, it is divided into two sub-categories. The primary subcategory is fixed probabilistic caching (FIX), and the secondary subcategory is dynamic probabilistic caching.

2.3.1 Fix-based Probabilistic Caching Strategies

This sort of content-deployment mechanism is based on the pre-determined probabilistic values to cache the consumer's interested content along the data delivery path [97, 120]. Caching of the content is taken individually at all routers without the involvement of collaboration among the nodes. FIX has no distinction regarding the transmitted content [121].

2.3.1.1 Everywhere Probabilistic Caching (EPC)

The Everywhere Probabilistic Caching (EPC) strategy lies in FIX and is based on probabilistic situations. According to probability, the demanded content is cached with probability at all the nodes that have empty caches along the downloading path. If the probability value is set to 1, then it works just like CCN default caching, that is, Leave Copy Everywhere (LCE) [122]. If it has a value other than 1, the disseminated content will be stored according to the probability value (e.g., probability based on strategy and the available cache).

Figure 2.4 shows the dissemination of two sets of content through different probabilities. If the caching is set to probability 1 at that moment, the content will be cached at all routers to facilitate the available cache on the delivery path. In Figure 2.4, Consumer A sends out an Interest1 to the network to retrieve content C1, but the desired content is not found in any router’s cache so Interest1 reaches the content provider.

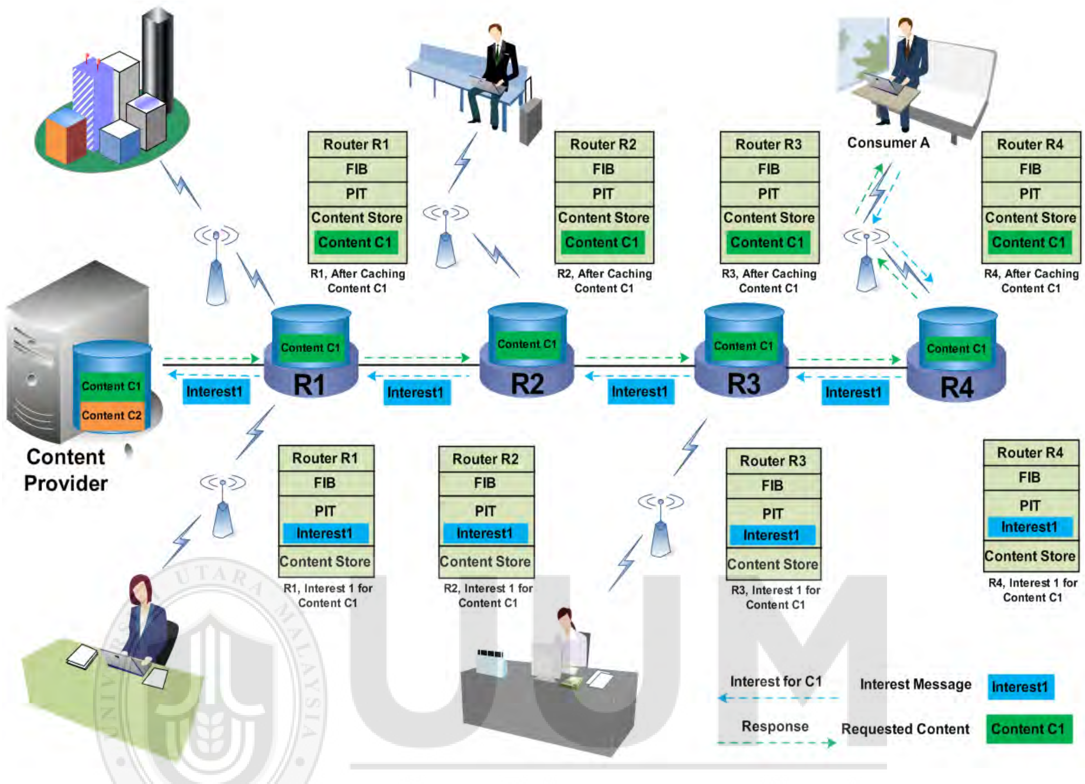

Figure 2.4. Everywhere Probabilistic Caching

Consequently, the provider responds to received Interest1 by sending the required content C1 to Consumer A, and a copy of C1 will be cached at all on-path routers (R1, R2, R3, and R4). Moreover, for the subsequent Interests, the C1 will be sent from the nearest router R4.

2.3.1.2 Priority-Based Probabilistic Caching (PBPC)

The Priority-Based Probabilistic Caching (PBPC) strategy lies in FIX and assigns the priorities to the content based on the stretch between the content providers and the consumers. Probabilities for the content vary at all nodes, and the disseminated sets of content are cached along the consumer's path based on their probabilistic value [14, 67]. The probabilistic value of the content is inversely proportional to the distance. Therefore, if the distance between the content provider and consumer is farther, the content will be indicated as a low probability to be cached along the data delivery path. On the other hand, if the distance between the provider and consumer is shorter the interested content will be posed as a high probability to cache the disseminated content along the data delivery path [122].

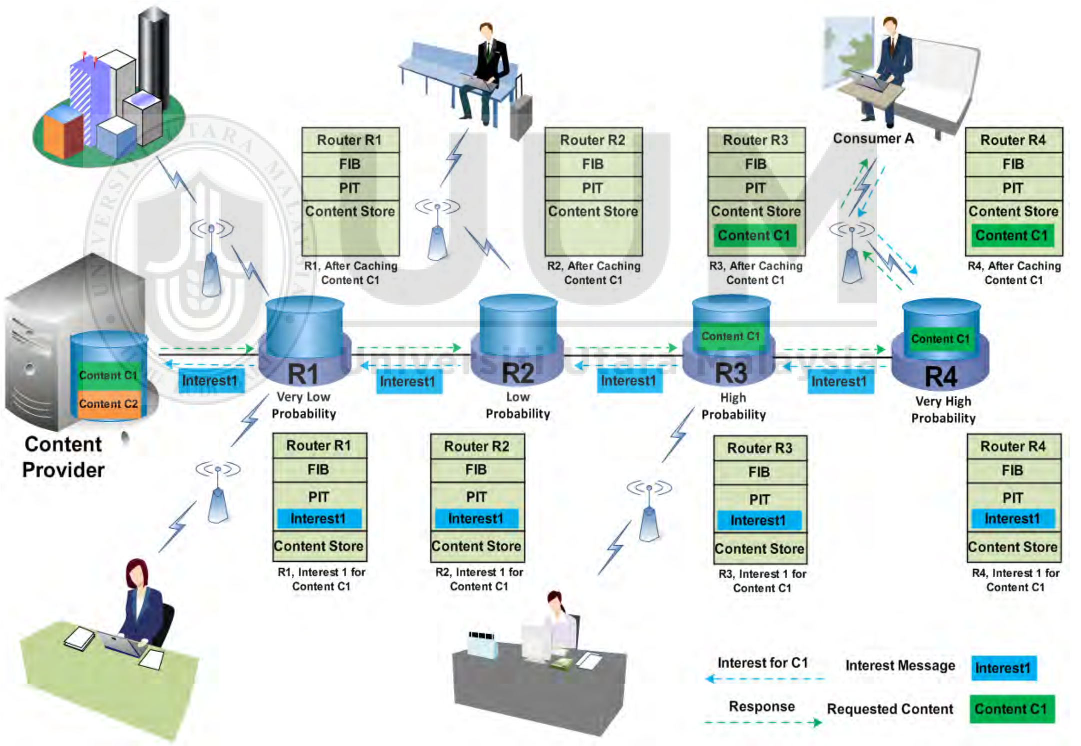

Figure 2.5. Priority-Based Probabilistic Caching

Figure 2.5 provides a visual of the entire process of PBPC, in which Consumer A sends an Interest1 to get Content C1, and C1 is found in the content provider's cache. As a result, the desired content is sent back to Consumer A, and a copy of C1 is locally cached along the delivery path. According to PCPB, the probability is divided to cache the disseminated content based on distance. If the distance is small, the router gets more chances to cache the content; however, as the distance increases, the probability decreases. In Figure 2.4, interested content C1 is cached only at routers R3 and R4 because R3 and R4 show the highest probability to cache C1 due to the shorter distance from Consumer A. On the other hand, routers R1 and R2 are located far from Consumer A, and both have low probability to cache C1 within their cache storage. Consequently, the content does not have much probability to cache at R1 and R2. Therefore, C1 only cached at R3 and R4 because of the short distance between Consumer A and these routers (R3 and R4).

2.3.1.3 Random-Based Probabilistic Caching (RPC)

Random-Based Probabilistic Caching (RPC) is derived from FIX, in which the caching decision of disseminated content is determined based on the randomly chosen probabilistic value [66]. RPC offers two advantages: (1) exhibits simplicity in structure, and (2) extracts low overhead when searching content for caching. On the other hand, RPC possesses low efficiency due to its uncertain and random nature. This mechanism is used as a benchmark to compare the performance of the latest establishing caching strategies [65]. In this strategy, a copy of the required content is cached at randomly selected locations along the data delivery path from provider to consumer.

RPC can be materialized into several kinds of approaches; for example, the caching decision can be performed based on the levels of tree structure topology. The caching depends on the algorithm regarding which levels of the caching router are available [123]. For instance, if level-1 is chosen for caching the transmitted content, then it will cache only at routers located in level-1 of the tree topology [104].

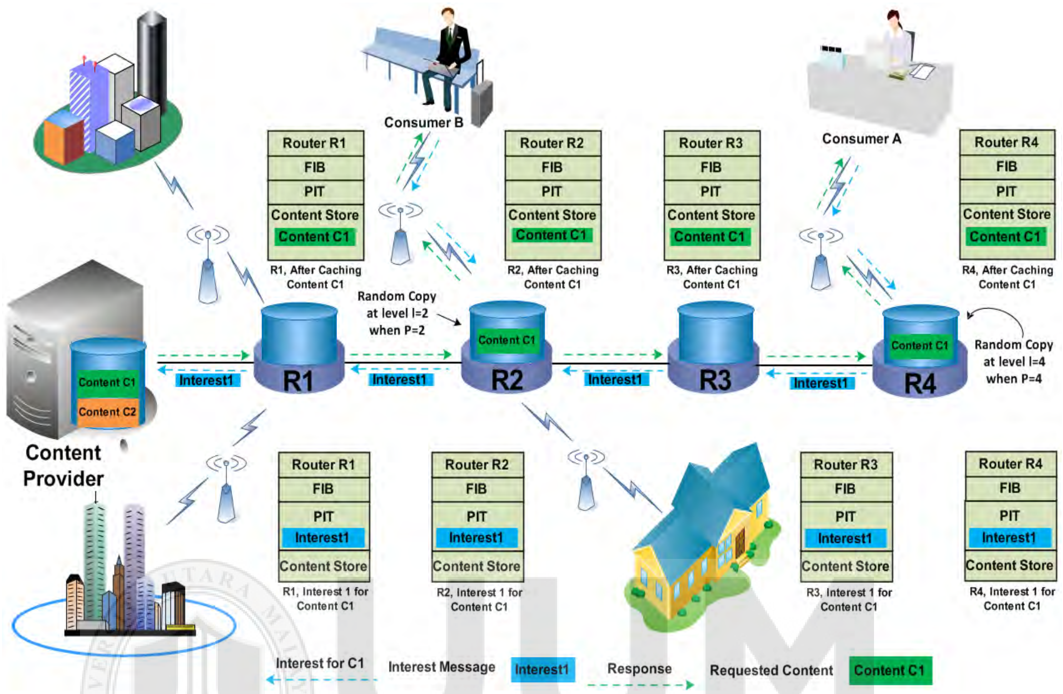

Figure 2.6. Random-Based Probabilistic Caching

On the other hand, if level-2 is selected for content caching, the content will be cached at the routers located at level-2 of the tree topology (see Figure 2.6), and when Consumer A sends out an Interest to retrieve content C1, C1 is cached at router R2. If level 4 is chosen, for example, the content will be cached at R4, as illustrated in Figure 2.6. The caching decision is predefined in the strategy algorithm, and it can be changed according to the nature of the environment.

Moreover, it can be changed on the basis of data traffic congestion and to reduce the distance between providers and consumers for subsequent Interests. RPC provides a simple structure and low overhead, but it becomes questionable due to the unpredictable nature of randomness and its different types of random structures. RPC does not provide a distinct position, which causes high retrieval times because there is a possibility of caching content far from the consumer, increasing the distance between providers and consumers.

2.3.2 Dynamic-probabilistic caching strategies

The focus of dynamic-probabilistic caching strategies is to enhance the structure of network topology with traffic patterns. The dynamic caching strategies are developed to diminish network and caching redundancy to mitigate the resource usage rates on the whole network. Caching decisions in dynamic probabilistic caching is dependent on the capacity of the routers' caches, the amount of data traffic, and the time required for content to be cached [59]. All content is incorporated with a time window to facilitate how much time they can be cached at a specific position; this interval is calculated via the Target Time window (Ttw) [98].

Moreover, two primitives, Time Since Inception (TSI) and Time Since Birth (TSB), are used for the calculation of path length. TSI is used within the Interest header, and TSB is assigned within the content header. When a consumer sends an Interest packet for some content, the value of TSI is initiated to 1, and the value increases by one when it passes through one hop. TSB is coupled with desired content. When interested content is hit at a provider, the value is initialized with 1, and this value is boosted up as the content traverses to the consumer. When a hit occurs, the TSI header of the Interest merges with content. However, its value remains unaffected as the content passes through the data routing path to the consumer [124]. TSI indicates the path length of the Interest from consumer to provider, and TSB represents the distance from provider to consumer [14].

2.3.2.1 ProbCache

ProbCache is a type of dynamic probabilistic caching [125] in which the caching decision is based on the cache capacity along with the data downloading path and the factor used to calculate the tendency to cache the content close to the consumer [126]. ProbCache aims to improve the caching tendency of the consumer's interested content to have it cached nearer to consumers. The probabilistic value for content caching along the data routing path can be calculated as given in the following:

$$\text{ProbCache} = \text{TimesIn} + \text{CacheWeight} \tag{2.1}$$

$$\text{TimesIn} = \frac{\sum_{n=1}^{x-y+1} Nn\ldots}{\text{Ttw}*\ \text{Nx}} \tag{2.2}$$

$$\text{CacheWeight} = \frac{\text{y}}{\text{x}} \tag{2.3}$$

where N_n indicates the cache storage of all the nodes along the path between source and consumer, N_x is the average cache capability of all the routers along the delivery path, Ttw is the Target Time window assigned by the mechanism that gives details about how much time content can be cached along the data routing path, and *n* shows the number of routers along the path. Moreover, x and y stands for the Interest header with TSI and interested content header with TSB, respectively.

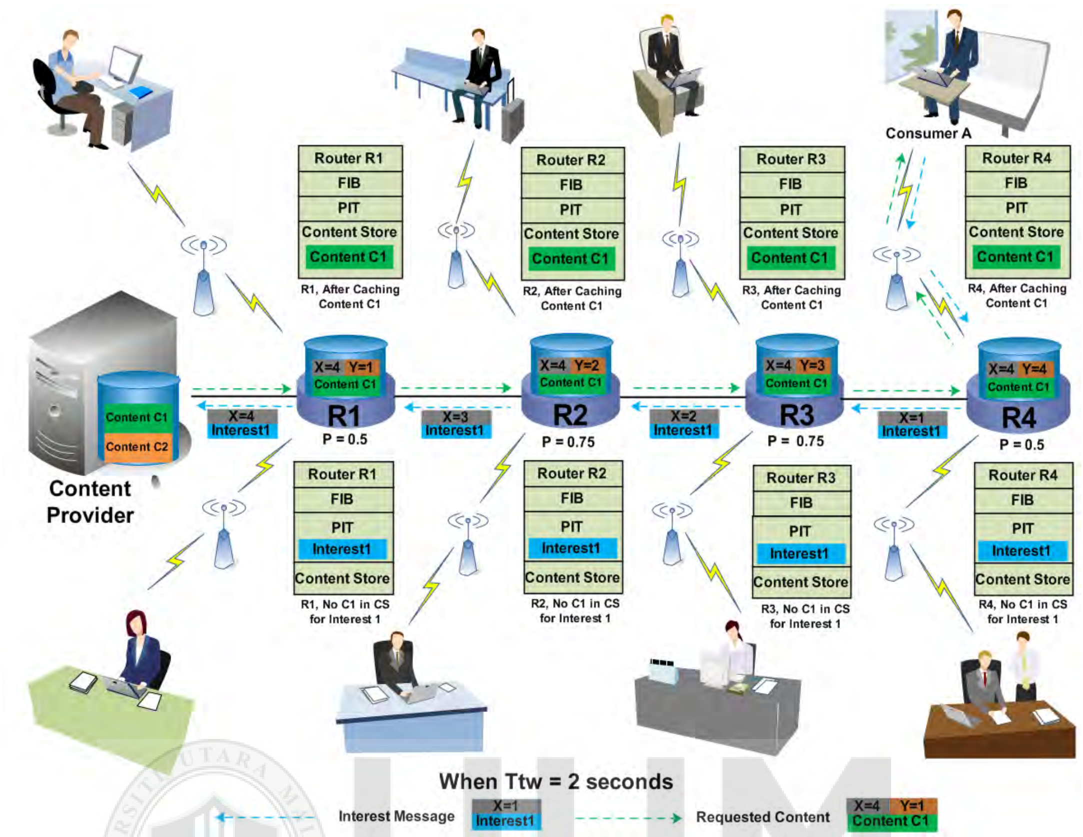

Figure 2.7. ProbCache

In Equation 2.2, y symbolizes the TSB header in which the number of nodes is calculated between provider and consumer, whereas x illustrates the TSI header coupled with the consumer Interest that quantifies the number of routers between consumer and provider. These equations need to be calculated at all router along the delivery path throughout content transmission [99]. However, the time period used to cache content at the subscriber path can be calculated by the Times-In factor. At the same time, the CacheWeight factor is used to cache content close to the consumer.

This caching strategy yielded poor performance for various caching spaces along the path. Figure 2.7 demonstrates the design of ProbCache: When Consumer A sends away an Interest to the content provider, at that moment, the value of the TSB header is incremented by 1 at each router. While the source acts in response to the Interest via sending interested content C1, at that time, the TSI value is attached along with the TSB header of the desired content. Although the TSI remains unchanged throughout content delivery, the value of TSB is raise incrementally at all nodes. If both values will be the same according to ProbCache, the interested content will be cached near Consumer A at the edge router R4.

According to ProbCache, the Ttw is quite small, which increases the content eviction rate and decreases the caching gain. When Consumer A throws an Interest for content C1, the value of x is added to at every router along the routing path from Consumer A to the provider. Subsequently, the provider responds to Consumer A by sending content C1; the value of x is attached within the content header, while the value of y is added to along the data routing path at all the routers (from provider to Consumer A). If the value of Ttw is agreed to as Ttw=2 sec for an identical cache size followed by calculating the TimesIn factor and CacheWeight factor, the probabilistic values would be equivalent to p = (0.5) R1, p = (0.75) R2, p = (0.75) R3, and p = (0.5) R4, alongside the path (where R1, R2, R3, and R4 are the caching routers alongside the subscriber path). Obviously, the probability is higher to cache content at the symmetric routers than at the edge routers [127]. Therefore, ProbCache dispenses a fair cache distribution along the delivery path.

2.3.2.2 ProbCache+

ProbCache+ [98] is the enhanced caching mechanism version of ProbCache that was developed to improve the unfair cache allocation among disseminated content. In this strategy, the cache allocation was enhanced by changing the value of Ttw along with the routers at the delivery path. To support this, an improved form of CacheWeight has been projected. According to ProbCache+, the value of Ttw is decreased by increasing the distance from consumers; in other words, Ttw is inversely proportional to distance [128]. The probability is amplified gradually along the delivery path as the content caches near the consumer and presents a high probability of maintaining its caching situation for a long time. The probabilistic value for content caching can be calculated as:

$$\text{ProbCache+} = \text{TimesIn} \times \text{CacheWeight} \tag{2.4}$$

$$\text{TimesIn} = \frac{\sum_{n=1}^{x-y+1} Nn \ldots}{\text{Ttw} * \text{Nx}} \tag{2.5}$$

$$\text{CacheWeight} = \left(\frac{\text{y}}{\text{x}}\right)^{x} \tag{2.6}$$

In these equations, Nn is the cache space of all the routers along the data routing path, and N_x is the average cache ability of the routers. Ttw is the Target Time window set down by the strategy regarding the length of time interval a content can be cached at the consumer's path, n is used to count the number of routers, and x and y symbolize the Interest header TSI and content header TSB, respectively.

To improve this caching, the CacheWeight Factor is modified in exponential form to calculate the probability of caching content at all on-path routers, as demonstrated in Figure 2.8. ProbCache+ enhances the ProbCache algorithm by distributing the probabilistic value according to the distance in hops from consumers; for example, for Consumer A, it will be (0, 0.32, 0.05, 0.5), along with router R1, R2, R3, and R4, respectively.

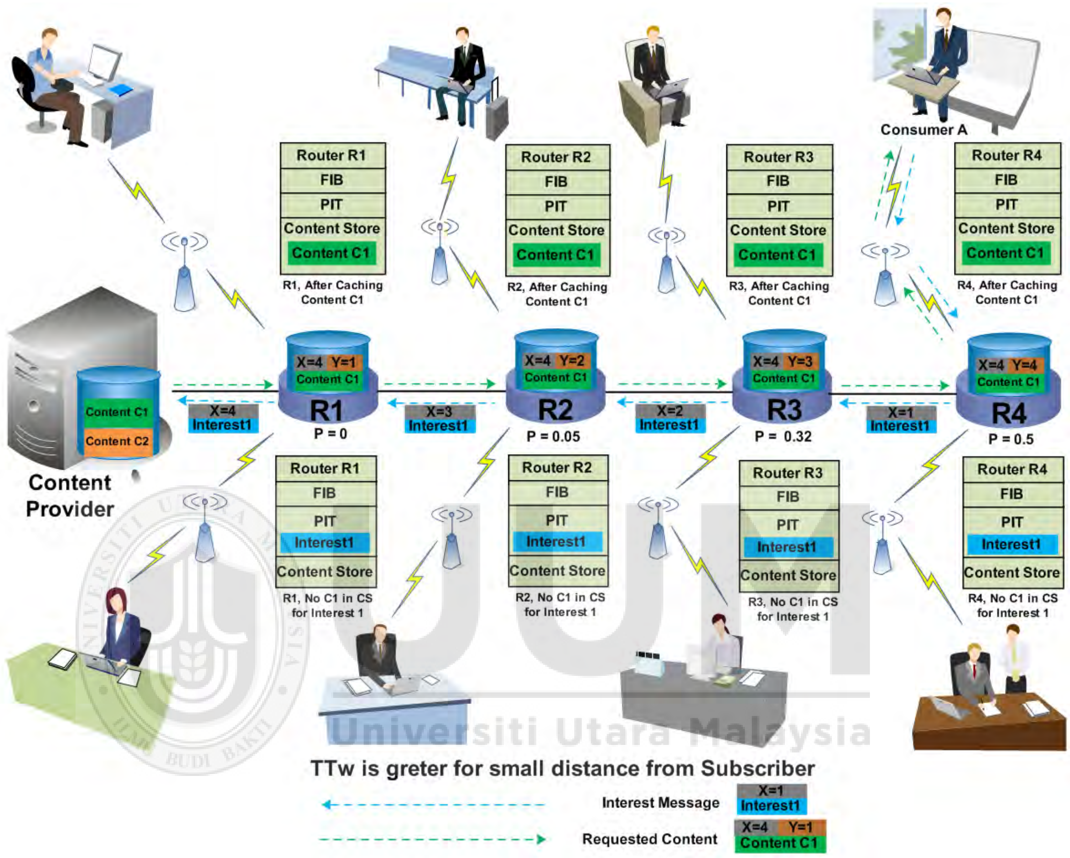

Figure 2.8. ProbCache+

This proves that the probability is rising toward the consumer and falling away from the provider. If the distance is long, the probability will be reduced. ProbCache and ProbCache+ aim to tackle the problems related to inefficient resource usage in different ways. They share the resources (cache) fairly among the content, but these strategies seem unable to achieve their goals completely.

Both caching strategies introduce a number of parameters that have no content distinction and need to be set arbitrarily. In addition, both strategies need to update the consumer Interests and content in the form of TSI/TSB, which increases the computational cost and memory consumption at all routers. Moreover, these caching strategies introduce Ttw by using a special variable that also increases the computational cost. Finally, both strategies do not distinguish cached sets of content and provider no criterion to handle the frequently interested content [118].

2.3.2.3 ProXCache

The ProXCache caching strategy has been derived recently from dynamic probabilistic caching to improve content redundancy and path redundancy. It implemented through a hypergraph, ProXCache strategy uses a specific header address combined with an Interest message as the consumer Interest for some content 0. Based on the proposed caching, the dissemination of consumer Interest packets a particular cache time interval, path computation, and hop distance are required. When a cache hit is recorded, the interested data content is thrown to the consumer, and the proxy location is attained by the publisher to cache a copy of interested content along the consumer's path as an augmentation of Leave Copy Down (LCD) (see Figure 2.9). ProXCache is different from the previous mechanisms due to the constant transmission of data delivered to consumers using cache time window, path distance, and hop distance. After that, the interested contents are cached probabilistically at the mid-routers of the consumer's path through a centrality computation algorithm.

Therefore, transmitted contents are cached at the proxy node and moved toward the consumer. Through PIT records, this operation is performed at the intermediate node.

In continuous Interests and network traffic, proxy positions, cache time windows, and path capacity of the network need to be obtained for each Interest. This caching mechanism was created to mitigate content redundancy and path redundancy;

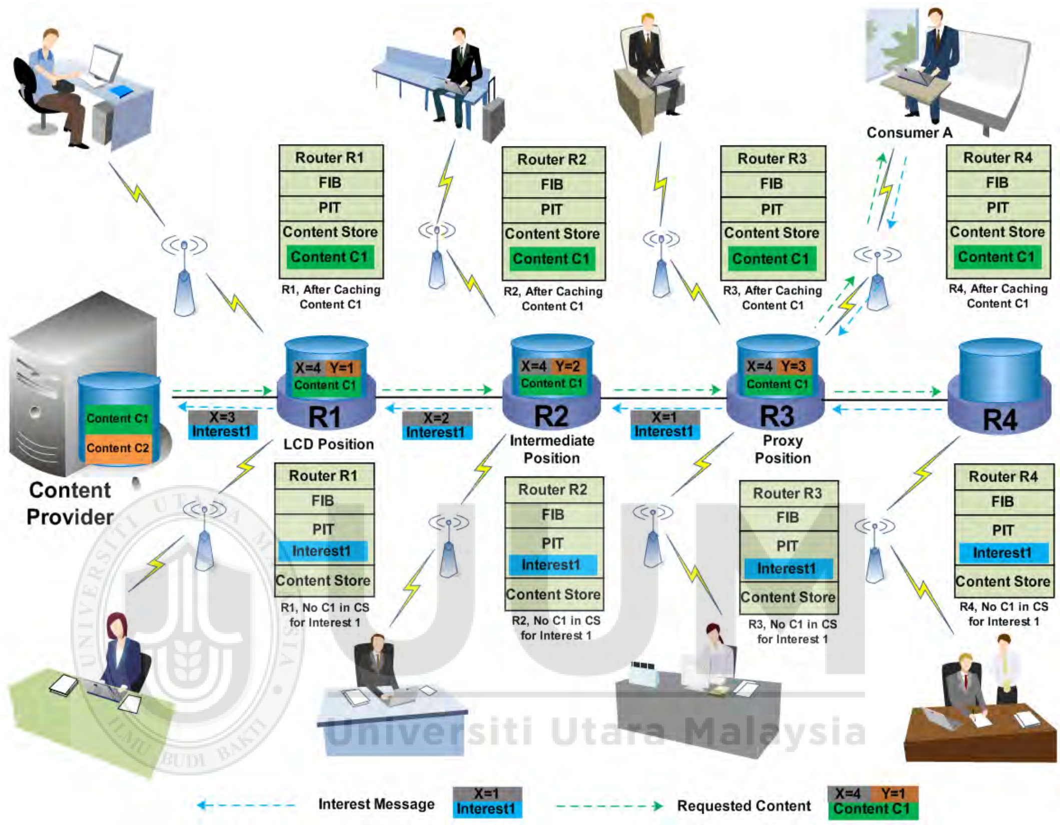

Figure 2.9. ProXCache

However, in short, stretch paths, it actually substantiates content redundancy. In addition, it increases the communication overhead because a number of variables need to be calculated during each transmission of Interest messages and interested content. ProXCache maximizes resource usage (caching) as millions of transmissions are required in a small time interval.

Figure 2.9 demonstrates the operation of the ProXCache caching strategy which clearly indicates the content redundancy along the short delivery path when Consumer A sends Interests messages. Consequently, the interested content C1 is cached at all the routers along the short stretch path by routers R1, R2, and R3. Moreover, ProXCache needs a number of parameters for the content dissemination which increases costs in term of resources and energy.

2.3.3 Max-Gain In-network Caching

MAGIC [18] was proposed to minimize the overall bandwidth consumption as well as the amount of caching operations. In MAGIC, content caching performs using two procedures. In the first procedure, content is cached in the CS and in the second procedure; the content replacement operation is performed to delete the content from the router's cache so that the new content can be accommodated. The caching operation is based on the popularity of contents and the hop reduction count. In MAGIC, two primitives, local cache gain, and Max-gain values are used to calculate the popularities of the requested contents. Each router locally calculates the local cache gain in which the numbers of placement and replacement operations are measured. MAGIC uses the local information (content name, Interest count, popularity count) related to contents to calculate the local gain. The local gain helps to find the appropriate node with location to cache the incoming contents along the data delivery path.

To find the router that has maximum local gain, the MAGIC adds extra information, named Max-local gain, and embeds it into the consumer Interest. When the Interest is generated for some content, the counter for Max-gain value is initialized to 0. As the Interest is received at any router, the MAGIC compares the values of Max-gain of Interest with a local gain of the router. If the local gain value is greater than the value of Max-gain, the router updates the Max-gain value. As the Interest along with its Max-gain reaches the content provider, the Max gain value is delivered to content and embedded into the content.

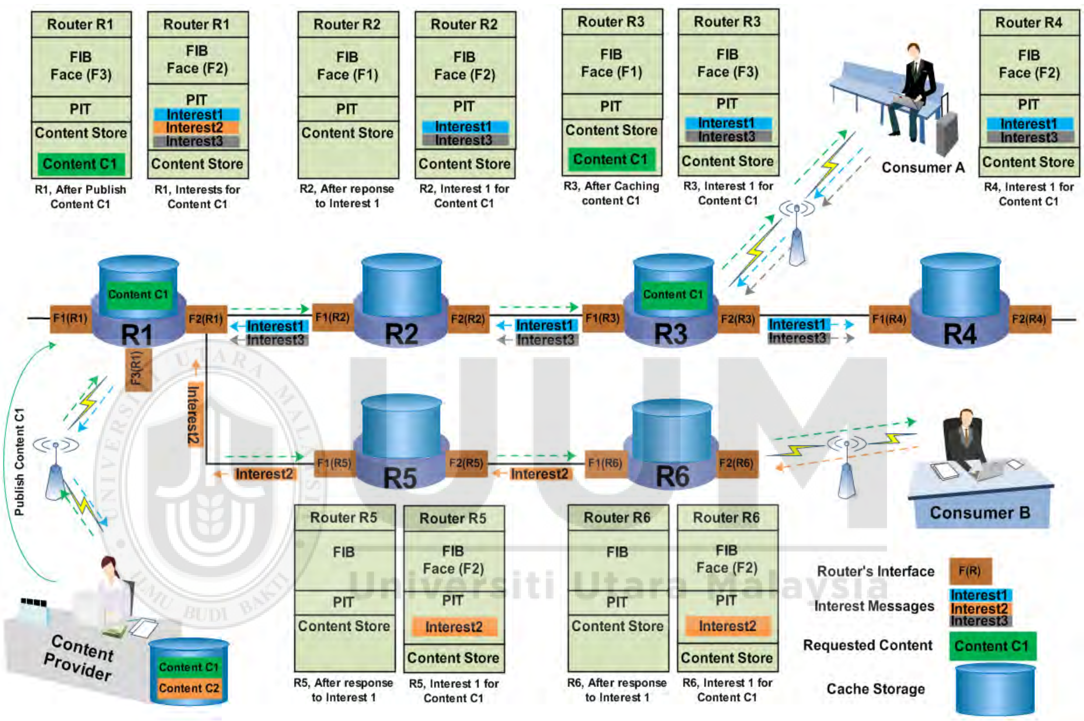

Figure 2.10. Max-Gain In-network Caching

Therefore, the content is cached along the delivery path if the Max-gain value of the transmitted content and the local gain value of the router are equal. Thus, the content with more Interests gets more chance to be cached at intermediate routers.

Figure 2.10 illustrates the content caching mechanism in MAGIC. In the given scenario, three Interests (Interest1, Interest2, and Interest3) are sent to retrieve the

Content C1 from router R1. In response to received Interests, the router R1 behaves as a provider and sends the required Content C1 to the consumers A and B. According to MAGIC, the local gain and Max-gain are calculated for all the received Interests. Therefore, the local gain and Max-gain are graters at router R4 as compared to other routers, so that the Content C1 is cached at router R4 which is near Consumer A because he sent more (two) Interests than Consumer B to download Content C1.

2.3.4 Hop-based Probabilistic Caching (HPC)

Hop-based Probabilistic Caching (HPC) [129] was proposed to overcome the issues of earlier probabilistic-based caching strategies. It was developed by using the enhanced version of $CacheWeight_y$ factor and $CacheWeight_{MRT}$. The $CacheWeight_y$ was enhanced to reduce the similar content replications and y parameter was used to determine the number of hops between a content provider and a consumer. $CacheWeight_y$ also helps to minimize the distance between the cached content and consumer by pushing the content towards the consumer to reduce the path length for subsequent Interests [130].

$$CacheWeight_y = \frac{1}{y+a}, \quad \alpha \geq 0 \tag{2.7}$$

$$CacheWeightMRT = MRTm + MRTexp \tag{2.8}$$

$$HPC = CacheWeight_y + CacheWeight_{MRT} \tag{2.9}$$

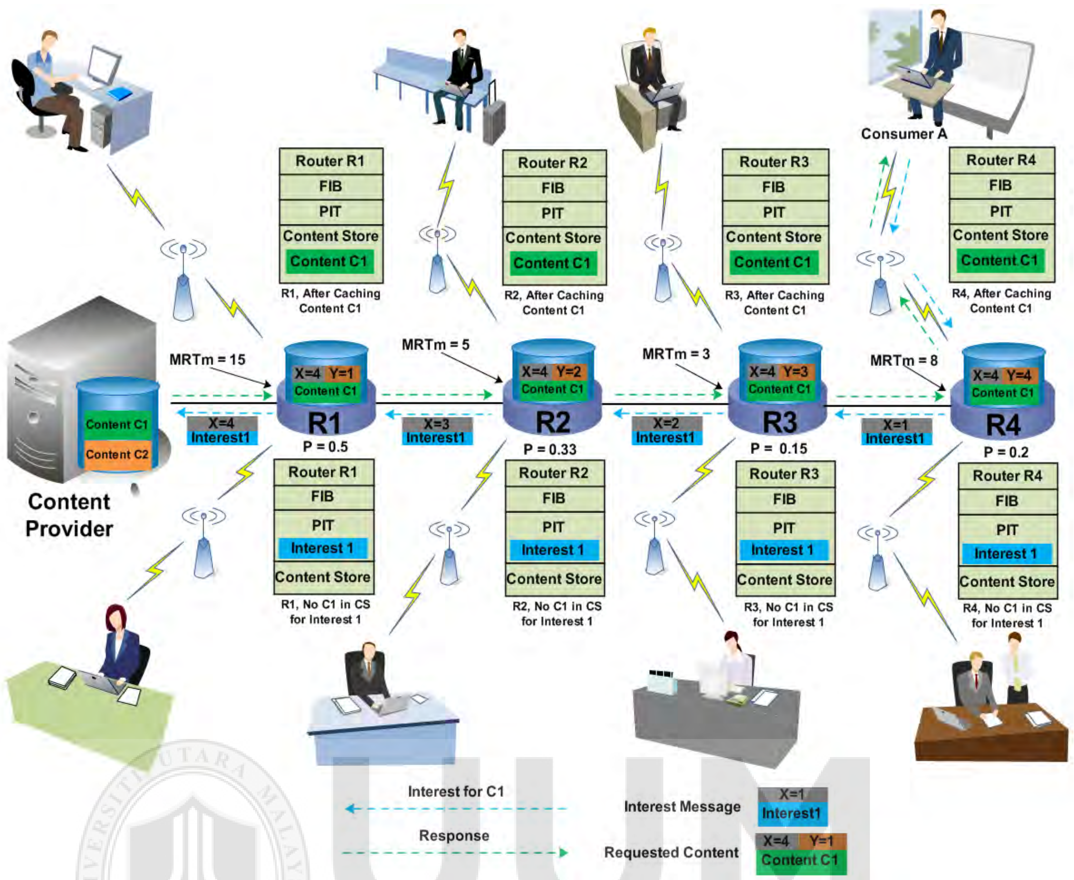

Figure 2.11. Hop-based Probabilistic Caching

The $CacheWeight_y$ factor is equal to the distance between the consumer and the provider. In equation 2.7, a show the distance in hop-count and it is used as a constant integer that assigns to the cache capacity of the whole path. The CacheWeightMRT refers to the duration that identifies for how long content can be cached along the data delivery path. CacheWeightMRT is derived using two factors as Mean Residence Time (MRT_m) and expected Mean Residence Time (MRT_{exp}). Suppose the value is selected as $MRT_{exp} = 5$ seconds, $a = 1$, and the value of the parameter is associated with 1, 2, 3, and 4 with routers R1, R2, R3, and R4, respectively. The value of MRT_m the factor is derived as 8, 3, 5, and 15 along the routers R1, R2, R3, and R4, respectively.

Therefore, CacheWeight_y factor is achieved as 0.5, 0.33, 0.15, 0.2 and this value is dispersed along the data routing path with each router as $R1 = 0.5, R2 = 0.33, R3 = 0.15$, and $R4 = 0.2$ as shown in Figure 2.11. In response to Interest1 (received from Consumer A), the Content C1 is cached at all the routers along the path according to the given probabilistic value.

2.3.5 Most Popular Cache (MPC)

In Most Popular Cache (MPC) [102], each node is associated with a special entity known as a popularity table, and three types of information are needed to store the cached content: the content name, the popularity count, and the threshold. All nodes need to calculate the number of incoming Interests for each content name to measure the popularity of content [63].

The threshold is the maximum value according to which the content is considered popular, and this value is fixed by the content caching mechanism. When the popularity count for a particular content name becomes equal to the threshold value, the content is labeled as popular. If a node holds the popular content, it recommends caching it at its neighbor nodes by sending a suggestion message. The suggestion message may or may not be acknowledged, depending on the resource's (i.e., the cache's) availability. When popular content is cached at neighbor nodes, its popularity is reinitiated to avoid flooding caused by the replication of analogous content.

Figure 2.12 elucidates the MPC caching mechanism in which two Interests are sent by consumers A and B to retrieve content C1 from router R4 and R6. At the same time, another Interest, Interest 1, is received from consumer C to retrieve content C2 from router R3.

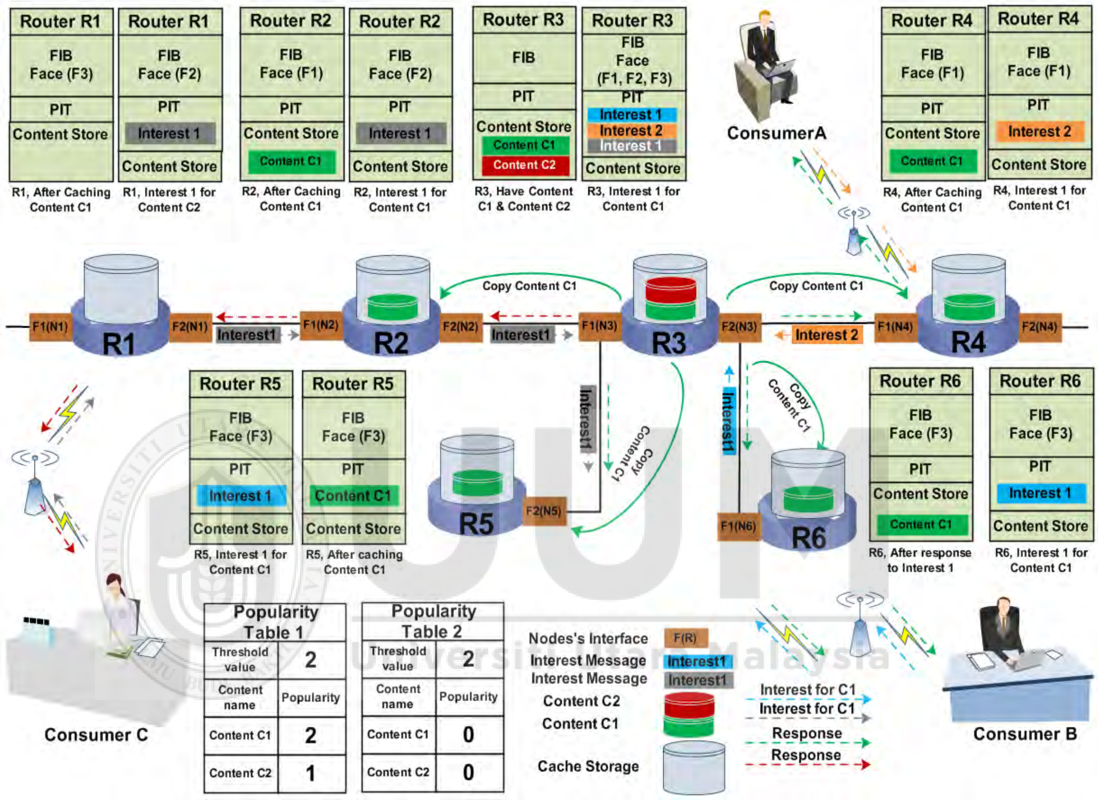

Figure 2.12. Most Popular Cache

According to MPC, content C1 is labeled as popular content because it received two Interests that were equal to the threshold value, as shown in Popularity Table 1 (see Figure 2.12). Therefore, router R2 sends suggestion messages to its neighbor routers (e.g., R2, R4, R4, and R6) to cache content C1, and the popularity of content C1 is reinitialized to 0, as shown in Popularity Table 2 (see Figure 2.12).

2.3.6 Cache Capacity Aware Caching (CCAC)

Cache Capacity Aware Caching strategy [19] was proposed by the composition of selective caching with cache-aware routing algorithm. It was designed to reduce loads of networks and servers. CCAC observes the recent cache consumption to estimate the available cache to accommodate new coming contents to be cached at a routing path. A node indicating highest cache capacity has more chances to cache the upcoming content within its cache. The cache-aware routing forwards the consumer's Interest to the suitable provider by using an extra feature as FIB. The cache capacity is calculated using an inverse function to the amount of recently used cache. Each node indicates a heterogeneous size of cache capacity. Therefore, the cache capacity value of the i^{th} node can be calculated as shown in followings.

$$CCV(i) = \frac{c}{L(i)} \times Cache_{size}(i) \qquad (2.10)$$

where c represents the compensation value in the case that the huge caching load, L, makes distortion among the network nodes. Two fields as CCV_i and Network Distance Value is integrated with each Interest to find the available cache and distance between a consumer and a content provider. As the hit is recorded, the content is forwarded to the consumer according to the following equation:

$$w_r = \frac{\log r}{log N_{total}} = \log_{N_{total}} r \qquad (2.11)$$

where N_{total} shows the total number of contents and r represents the ranking of popular contents. The popularity of a content is measured by taking the sum of Interests generated for particular content. Therefore, a weight is associated with each popular content that helps the content to be cached with higher possibility. The CCV_i is attached to the content as the provider makes a reply to the Interest and threshold is individually determined by combining the CCV_i and w_r at all nodes along the routing path.

$$CCV_{th} = CCV_{highest} \times w_r \tag{2.12}$$

where CCV_{th} represents a threshold, which is used to make content popular.

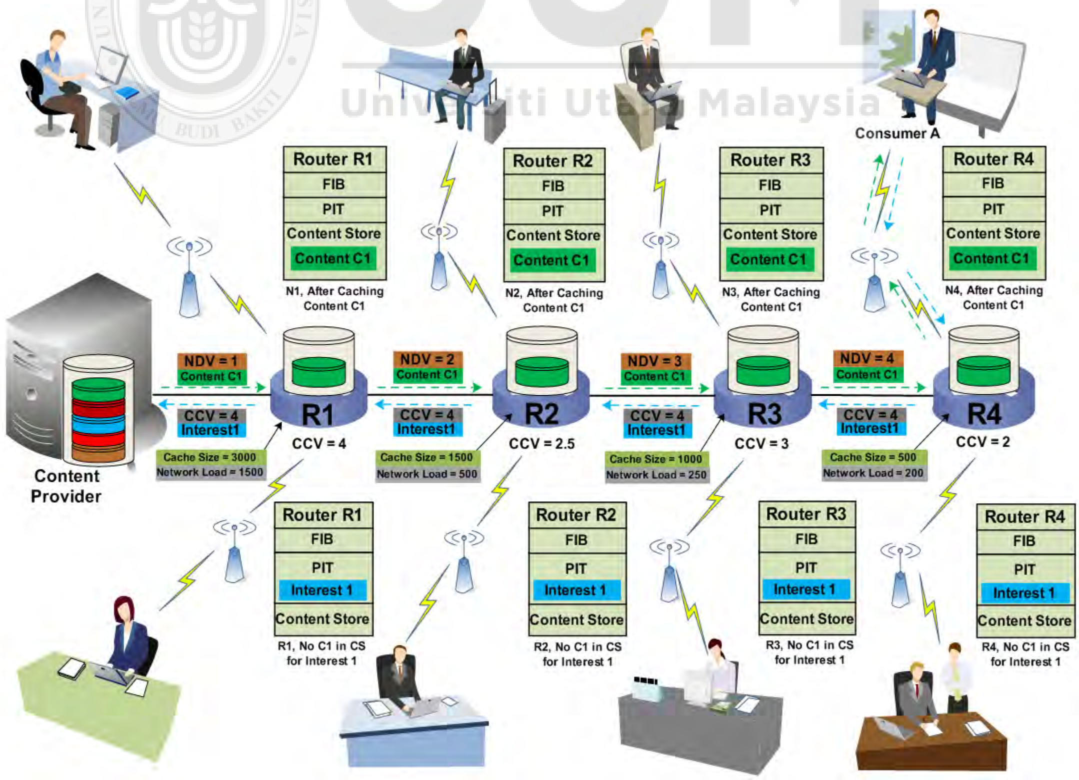

Figure 2.13. Cache Capacity Aware Caching

In Figure 2.13, the content caching mechanism of CCAC is illustrated in which consumer A sends an Interests to retrieve the content C1. As a hit is recorded, a content provider sends the requested content C1 to consumer A, and a copy of C1 is cached at intermediate router according to the given algorithm during its dissemination from Content Provider to consumer A.

2.3.7 Leaf Popular Down (LeafPopDown)

LeafPopDown [36] is a popularity-based cache management strategy used to manage the cache of nodes in NDN. LeafPopDown caches popular content at the downstream router and edge router among the data downloading path. When a consumer sends an Interest for content, a path is created from the consumer to the provider node for further communication. As the cache hit occurs, the provider node adds an entry in its popularity table. As the entry for that particular content exceeds the threshold (value to identify the popular content) the content is suggested as popular. The popular content is sent back to the interested consumer, and a copy of the content is cached at a neighbor node, which is placed one level down from the provider node, and one copy is cached at an edge node located just before the interested consumer.

In LeafPopDown, the highly interesting content (i.e., popular content) is gradually cached close to the consumer [37]. Figure 2.14 demonstrates the LeafPopDown caching strategy. In this figure, consumer A sends several Interests, to retrieve content C1. After several Interests are received for C1, it is selected as popular.

Therefore, router R1 behaves as a provider and sends required content, C1 toward consumer A, and a copy of C1 is cached at downstream router R2 and edge router R6 during its transmission from provider to consumer. When consumer B sends a number of Interests to retrieve C1 and, thus, C1 is selected again as popular, router R2 becomes a provider and pushes C1 toward consumer B, and a copy of C1 is cached at downward router R3 and edge router R4.

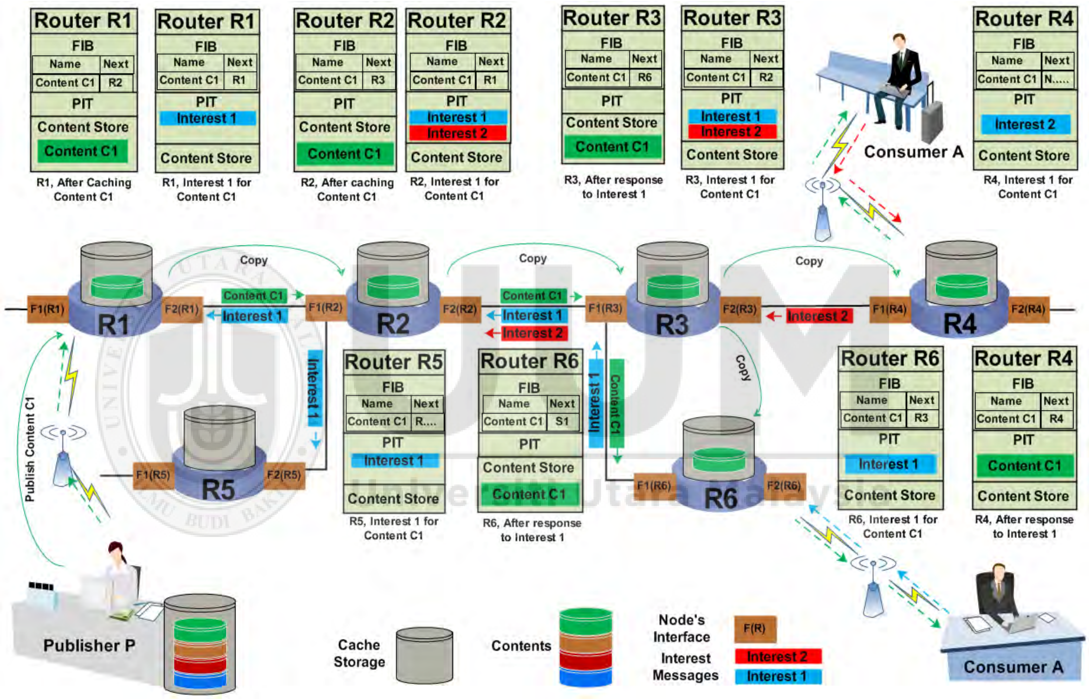

Figure 2.14. Leaf Popular Down

2.3.8 WAVE: Popularity-based Caching Strategy

This caching strategy is designed to achieve data distribution in chunks. The objective of the WAVE strategy is the efficient distribution of contents as well as reducing the average cache management cost. In WAVE strategy, content is cached in the form of chunks on the basis of the requirements of a consumer. All the chunks are associated with a relation known as inter-chunk relation. The WAVE strategy modifies the number of chunks imparted to a cache according to the popularity of the content (the number of Interests received by a specific content). The WAVE strategy exponentially increases the number of chunks to be cached and gradually forwards the chunks towards the consumers.

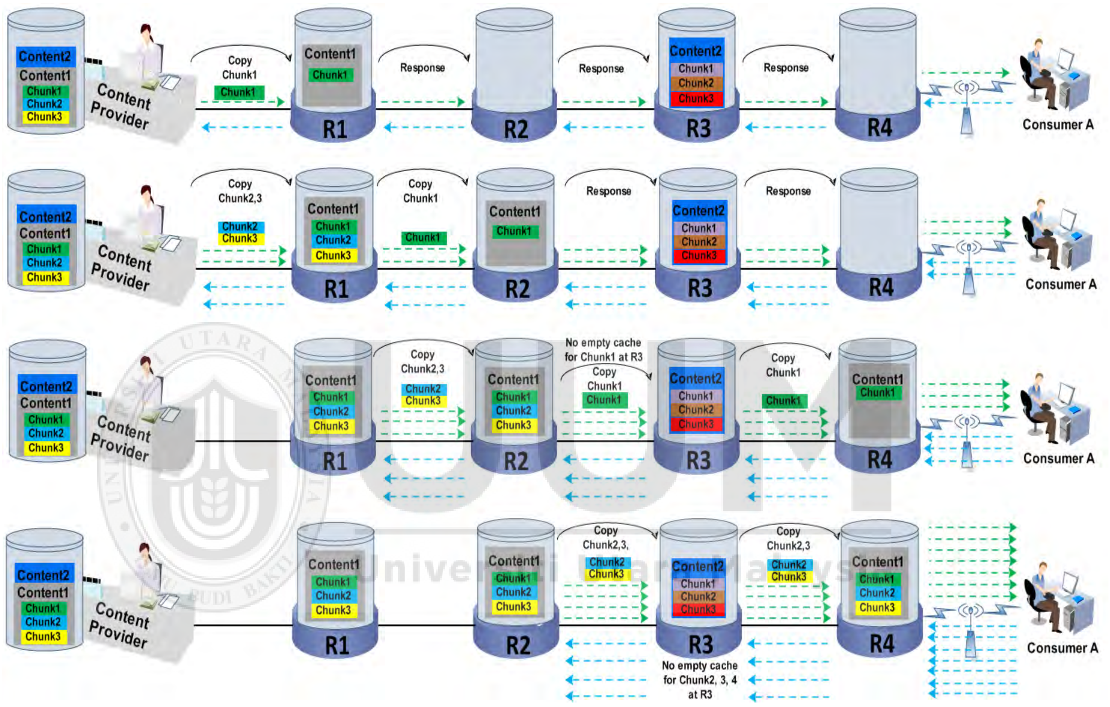

Figure 2.15. WAVE: Popularity-based Caching Strategy

All the routers individually decide whether the received chunk has to be cached or not. In this strategy, the upstream routers recommend to its down-stream routers to cache the coming chunk and this recommendation can be ignored by the router if the cache space of the down-stream router is full. If the downstream router does not have cache space to store the coming chunk, the chunk will be stored at the subsequent down-stream router. When the content provider receives an Interest for specific content or file, it divides the content into chunks to result in the efficient data distribution. For instance, the requested content is divided into 50 chunks and the provider will send that content in the form of chunks to the consumer. All the chunks of content have a specific type of index and the index is used in collaboration with the routers to avoid caching of disorganized chunks because each router individually decides regarding the caching operation. Moreover, a cache flag bit (e.g., a data packet header) is associated with each chunk for organized data delivery.

Figure 2.15 illustrates the content caching mechanism using the WAVE. From the given scenario, the Content Provider receives Interest1 that is generated by consumer A for retrieving content C1. In response to the interest, the Content Provider sends content C1 and Chunk1 is cached at router R1 during the first transmission of content C1 from the provider to the consumer. After a while, Interest2 is sent by consumer A to download content C1 again. Consequently, content C1 is sent to consumer A and the chunks (Chunk2, Chunk3) belonging to C1 are exponentially cached at router R1 and Chunk1 is cached at router R2 in the second transmission of content C1. This process is continuously performed, and the chunks are gradually forwarded towards the consumers as shown in the given scenario (Figure 2.15). In response to Interest4, the entire chunk reaches the edge router R4 and all the subsequent Interests will be accomplished from router R4.

2.4 Critical Analysis

The management of the cache depends on how the cache is deployed and managed by the network. The effective management of the cache demonstrates a mechanism that places a copy of the transmitted content into the network caching system efficiently [14]. This mechanism reduces the server stretch length for subsequent Interests. Further, it increases the cache hit rate and availability of data [15]. Therefore, the intention of on-path caching is to reduce the usage of the resources of the Internet. For this purpose, on-path caching delivers several cache management strategies that help in overcoming the data transmission problems such as content redundancy and path length between the publisher and the consumer. For efficient caching distribution, several placement strategies were proposed. These strategies were developed to implement the idea of popular content in which the content that receives more Interests indicates higher popularity and is cached along the data delivery path.

Table 1 illustrates certain common features and challenges of the diverse on-path caching strategies. All the strategies are analyzed on the basis of popular metrics that are used to evaluate the on-path caching performance. The common metrics are cache hit ratio, content diversity ratio, content redundancy ratio, stretch ratio.

The probabilistic caching received a high interest from researchers because of its multiple content selection mechanisms. It provides several ways to cache content during their transmission to fulfill the subsequent requirements for consumer Interests. Basically, the probabilistic caching is divided into two categories in which the fixed-based probabilistic caching plays an important role to enhance the overall caching performance.

The EPC caching mechanism was proposed to increase data availability near the consumers [131]. This mechanism supports the redundancy that increases the utilization of resources unnecessarily that caused the large number of caching and eviction operations and reduce the cache space for incoming content, due to which the stretch is maximized because new Interests need to forward to the main source due to less priority to cache most popular contents nearest the interested consumers. Moreover, it supports the content homogeneity that decreases the number of diverse contents within the cache of the node and consequently the redundancy is exploiting. The PBPC decreases the distance for the subsequent Interests that results in high performance in terms of caching gain, stretch, diversity, and redundancy, but some issues are still exists in this caching structure, For example, the content will cache at all the router alongside the data downloading path when the path length is very small, which maximize the redundancy, usage of resources (i.e., cache) and it will reduce diversity as well as cache hit when the cache is overflow.

The RPC was proposed to reduce the path length between the consumer and the intermediate cached copy of requested content. This technique delivers a high cache hit ratio and low latency rate. However, it demonstrates a lack of efficient caching for popular content. Moreover, it increases redundant content duplication and therefore reduces the content diversity. In addition, the RPC increases memory consumption by caching similar content at multiple locations. At certain times, it performs better in terms of the cache hit ratio and stretch ratio; however, this depends on the free cache space along the data delivery path. It does not define any criteria for popular content to be cached at intermediate locations. All the fixed based probabilistic caching strategies have limited usage of resources and deliver fast propagation of content.

ProbCache provides a fair allocation of resources along the delivery path amongst disseminated contents but it has no any content distinction, updating of Interest and content reply packets (computational overhead) arbitrary description of parameters [118]. This strategy does not succeed to provide any compensation which proves that it does not get done its goals [97]. Both fixed and dynamic probabilistic caching endeavor to diminish the content redundancy but the redundancy still exists. The dynamic probabilistic strategies established a new attribute Ttw which is used to determine a particular time interval to cache content alongside the data path but it reinforces the content eviction rate. Fixed probabilistic caching strategies have low overhead but there is no cooperation amongst the nodes which maximize the redundancy and enlarge the consumption of the resources. Random probabilistic cache scheme introduced simplicity and low overhead but it does not have any content as well as position distinction and it delivers unpredictable nature [65, 66]. ProbCache and ProbCache+ aim to tackle the problems related to the inefficient usage of the resources in a different way. They share the resources (cache) fairly among the contents but these strategies seem to be unable to achieve their goals completely [125].

Both caching strategies introduce a number of parameters that have no content distinction and need to set these parameters arbitrarily. Both strategies additionally need to update the consumer Interests and contents in the form of TSI and TSB that increase the computational cost and memory consumption within all routers. Moreover, these caching strategies introduce Ttw by using a special variable that also increases the computational cost. In addition, these strategies have no distinction about the cached contents and there is no criterion to handle the frequently Interested contents [130]. The hope-based caching was developed to enhance the performance of edge caching by pushing the contents near to the consumers and cache it for a particular time. ProXCache caching scheme has benefits related to the useful information and this mechanism works like a server that prevents the network from harm data [28]. It reduces the content redundancy and increases content diversity through implementation of proxy positions. Moreover, this machine is capable to reduce the caching and eviction operations that minimize the resource consumption and increase the usability of popular contents.

MAGIC was established to reduce bandwidth consumption and stretch (hop reduction). Moreover, it also provides a solution for the problems of least recently used contents in the different strategies. While caching contents locally, the best possible position of caches is required. It was concluded from the existing study that there is a basic requirement of a caching strategy that will place the transmitted contents at the optimal routers, which will minimize the average cost, usage of resources, bandwidth consumption, and manage the cache more efficiently. MAGIC demonstrates a high cost and requires more resources to compute the max gain and local gain values; consequently, the content retrieval time is increased, and the cache hit ratio is reduced.

HPC introduced a new factor MRT that is used to specify the time to cache a content along the path which increases the usage of the resources. Although, mean values do not establish any metric in that case if the distribution function of the contents is neither normal nor uniform [98]. It also keeps the high level of content redundancy that increases the amount of homogeneous caching operation that results the diversity is reduced. Moreover, similar caching operations reduce the hit rate due to the small amount of cache space because it does not accommodate the diverse number of content. It associates a specific time span to cache all the content during their dissemination from the provider to the consumer. The caching duration increases with decreasing distance from the desired consumer [65, 66]. Similarly, the HPC also improves the homogeneous content replication at multiple locations.

The MPC was introduced to promote the most downloaded (popular) content. In this way, the MPC received a higher cache hit ratio. However, the MPC utilizes extra space in the cache because it generates popularity tables for all the contents and millions of entries for similar content names and their popularity counts are cached in the popularity tables [132]. The MPC generates homogeneous replications of the content by caching the same content at all neighbor nodes. If the content provider and the consumer are associated with a small-stretch path, the MPC maintains diversity at its minimum level because of the multiple redundant caching operations. Therefore, the diversity ratio cannot satisfy the efficiency.

However, if the content becomes popular again, there is no other distinction to stop the flooding of similar content in the network, which increases the utility of the resource (cache). Moreover, the caching of similar content at multiple locations increases the usage of cache storage to manage the popularity table for each content name at all routers, which increases the communication and searching overhead that is used to decide which popular content will be cached at the neighboring routers; moreover, multiple contents indicating the same popularity are also stored within the limited cache size. In addition, the MPC is not focused on the time consumed to calculate the popularity of the content. Moreover, there is a lack of criteria in the MPC to manage the most recently used content because this content will remain unpopular according to the caching nature of MPC.

CCAC claims to provide content caching services with content routing services. For both services, it uses several entities such as CCV_i, CCV_{th}, additional FIB, w_r (content ranking), and popularity for contents. All these entities execute at all the routers separately whenever an Interest is generated, or a Data is transmitted to the consumers. In this way, the communication overhead is increased and consequently, the cache hit ratio keeps at its minimum level because millions of Interests are generated and correspondingly contents are transmitted in a very short interval. Moreover, CCAC shows the content redundancy because it distributes cache capacity along the data routing path within all nodes and all the contents are replicated wherever they found cache space. Similar types of popular contents are cached at multiple nodes that decrease the overall diversity ratio.

The LeafPopDown was proposed to reduce the content redundancy. However, in the LeafPopDown strategy, the content caching mechanism increases the number of homogeneous content replications, which reduces the amount of cache storage that is required to accommodate the large amounts of diverse content. It also increases the resource (cache) utilization and bandwidth consumption by caching analogous content at different locations. The LeafPoPDown does not define any criterion for caching the less-popular content, which increases redundant content at multiple locations; thus, the overall content redundancy is maximized. In particular, it decreases the content diversity because it increases the amount of identical content within the network cache, which reduces the amount of memory space available for less-popular content. Consequently, most of the Interests for less-popular contents have to be forwarded to the remote content providers to be satisfied.

Cho et al. designed WAVE [75] popularity-based caching strategy is designed to achieve data distribution in chunks. The objective of the WAVE strategy is the efficient distribution of contents as well as reducing the average cache management cost. In WAVE strategy, content is cached in the form of chunks on the basis of the requirements of a consumer. All the chunks are associated with a relation known as inter-chunk relation. The WAVE strategy modifies the number of chunks imparted to a cache according to the popularity of the content (the number of Interests received by a specific content). The WAVE strategy exponentially increases the number of chunks to be cached and gradually forwards the chunks towards the consumers. All the routers individually decide whether the received chunk has to be cached or not. In this strategy, the upstream routers recommend to its downstream routers to cache the coming chunk and this recommendation can be ignored by the router if the cache space of the downstream router is full. If the downstream router does not have cache space to store the coming chunk, the chunk will be stored at the subsequent downstream router. When the content provider receives an Interest for specific content or file, it divides the content into chunks to result in the efficient data distribution. For instance, the requested content is divided into 50 chunks and the provider will send that content in the form of chunks to the consumer. All the chunks of content have a specific type of index and the index is used in collaboration with the routers to avoid caching of disorganized chunks because each router individually takes the decision regarding the caching operation. Moreover, a cache flag bit (e.g., a data packet header) is associated with each chunk for organized data delivery.

The popularity is measured by taking the sum of Interests received for a particular content name and set the threshold to evaluate the next popular contents. The HPC, WAVE, CCAC, and MAGIC select the popular content that has received maximum number of Interests recently and this maximum number of Interests becomes the threshold value. Therefore, only the content is selected as popular that has received the maximum number of Interests. However, if a large number of contents are located far from the desired consumers and numerous Interests are received for those contents, but the popularities of these contents are slightly lower as compared to the selected popular content and hence, the contents will not be suggested as popular to be cached at intermediate nodes.

As a result, all the coming Interests for those contents need to traverse several hops to find the required contents and thus the overall stretch ratio will be increased. However, some popularity-based caching strategies such as MPC and LeafPopDown have a very small threshold that increases the redundant caching operations by less popular contents and reduces the chance to accommodate the highly popular contents near the consumers [65, 66]. Consequently, the cache hit ratio is reduced because frequent caching operations are occurred by the less popular contents. Thus, all these strategies focus on the popular or most recently interested contents that cannot fulfill the efficient threshold and content caching requirements for the most popular and

Table 2.1.
Caching Strategies contributions and challenges

Caching Strategies	Aim/Goal	Strength	Weaknesses
Everywhere Probabilistic Caching	Reduce the content redundancy High cache hit	• Simplicity • Low overhead	• No popularity distinction • No position distinction • Favors low-stretch delivery paths Cache redundancy on short-stretch
Priority-based Probabilistic Caching	Reduce the distance from the source	• Average cost	• No popularity distinction • No position distinction • Cache redundancy on short-stretch
Random-Probabilistic Caching	Randomly cache content	• Simplicity Low overhead • Reduce delay	• Unwanted cache • No popularity distinction • No content distinction • No position distinction • Unpredictable random nature
ProbCache	Fair allocation of capacity resources on a delivery path among contents	• Fair allocation of capacity resources	• No popularity distinction • Update of content request/content reply packets • Unfair allocation of cache capacity • (an unfulfilled goal) • Arbitrary definition of parameters
ProbCache+	Fair allocation of capacity resources on a delivery path among contents	• Fair allocation of capacity resources	• No popularity distinction • No content distinction • Update of content request/content reply packets (computational overhead) • Arbitrary definition of parameters
ProXCache	Deploy proxy router	• Less redundancy	• No popularity distinction • No content distinction • Arbitrary definition of parameters
MAx-Gain In-network Caching (MAGIC)	Minimize bandwidth and cache popular content	• Low bandwidth • Less memory consumption	• High threshold value • High delay • Low cache hit ratio • Diversity ratio
Hop-based Probabilistic Caching (HPC)	Pushes content towards the consumer for a particular time span	• High cache hit ratio • Short stretch ratio	• High threshold value • High redundancy ratio • Less diversity ratio • High bandwidth • High memory consumption
Most Popular Cache	Cache popular content along the data routing path	• High cache hit ratio • Low stretch ratio	• Low threshold value • High redundancy ratio • Less diversity ratio • High replication of same content memory
Cache Capacity Aware Caching (CCAC)	Cache popular content along the data routing path	• Low bandwidth • Short stretch	• High threshold value • High redundancy ratio • Less diversity ratio • High replication of same content memory • Less hit ratio
LeafPopDown Cache	Cache popular content at edge and downstream router during content dissemination	• High cache hit ratio • Short stretch ratio	• Low threshold value • Less diversity ratio • High redundancy ratio • High memory consumption
WAVE Popularity-based Caching	Cache popular content at all nodes	• Chunk level caching	• High threshold value • Less diversity ratio • High redundancy • Less stretch ratio • Low cache hit ratio

least popular contents that caused multiple homogeneous contents' replications along the publisher-subscriber path. Consequently, the overall caching performance is reduced in terms of caching hit ratio, stretch ratio, content redundancy, and diversity ratio. Therefore, all these strategies focus on the most popular or most recently interested content that cannot fulfill the caching requirements for less popular contents. Consequently, the overall caching performance is disturbed in terms of caching hit ratio, stretch, redundancy, and diversity.

After the critical analysis of the state of the art caching strategies, a new dynamic popularity-based cache deployment strategy that will enhance the threshold value as well as the content caching positions along the publish-subscribe path. As a result, the overall caching performance will be improver in terms to get efficient cache hit ratio, content redundancy, content diversity, and stretch ratio. However, the proposed caching mechanism in this study is more flexible to handle these issues of caching. The proposed caching strategy not only prefers the most frequently interested content but also handles the less popular contents to increase the overall caching performance.

2.5 Summary

In this chapter, the NDN architecture is explained. NDN caching obtained great attention from the NDN research community. The research community is analyzed that the cache management is a difficult task because the implementation of caching is a critical issue. The main issue is how we can use the cache of a router in a more efficient and well-organized manner. This issue occurs due to inefficient distribution of information. Therefore, a number of deployment strategies are recently built for

NDN to make distribution of information in a well-organized approach. Many deployment strategies are explained to understand the literature and related study. All these strategies are still facing some critical issues. Therefore, effective caching still needs to implement an efficient cache deployment strategy to achieve maximum efficiency. Therefore, a new dynamic popularity based caching strategy is proposed after taking the critical analysis of the earlier caching strategies to enhance the overall caching performance.

CHAPTER THREE
RESEARCH METHODOLOGY

This chapter will present the complete information about the design of the proposed strategy to manage the cache of Named Data Networking (NDN). According to different requirements, a number of deployment strategies have recently been proposed using diverse caching natures such as; according to cache availability, cache capacity, and content popularity. The aim of this research is to design a content deployment strategy for NDN named as Compound Popular Content Caching Strategy (CPCCS). It is used to store disseminated contents with router's cache during their transmission from provider to consumers. However, the CPCCS approach increases the availability of cache to accommodate desired contents to provide a fast delivery service as compared to the existing caching strategies. As such, it requires a methodology to understand each step to achieve the proposed objectives. As a result, this study adopts the Design Research Methodology (DRM) to introduce the main stages of research according to the nature of DRM.

The chapter starts with the overall research approach as presented in Section 3.1. Section 3.2 introduces the first stage of DRM named Research Clarification (RC) that discusses the purposes of the RC stage, methods to support this stage and the main deliverables. Furthermore, section 3.3 describes the second stage as Descriptive Study-I (DS-I) that presents the particular steps to obtain the efficient understanding of the present situation, designs a reference model and proposes a conceptual model for proposed strategy (CPCCS). Section 3.4 highlights the methods adopted in designing the proposed CPCCS mechanism in line with the third stage of DRM, named and this stage named as Prescriptive Study (PS). Section 3.5 is associated with the fourth stage of DRM named as Descriptive Study-II (DS-II) in which, methods for performance evaluation and its metrics are discussed. Finally, the chapter is summarized in section 3.6.

3.1 Research Approach

The aim of this research is to design a cache deployment strategy to enhance NDN cache performance in terms of the cache hit ratio, content diversity, stretch ratio, and content redundancy as well as improve the efficiency of the data transmission and it will be implemented by considering the other content deployment strategies of NDN. This demands a careful mapping between effective management of cache to place an efficient distribution of contents to reduce the server load along with low response time.

In order to select a suitable method, DRM provides diverse methods and combination of methods for research. Moreover, it offers step by step guidelines for difficult research and provides systematic planning. DRM consists of four stages as;

- Research Clarification (RC)
- Descriptive Study-I (DS-I)
- Prescriptive Study (PS)
- Descriptive Study-II (DS-II)

The DRM stages are described in the following sections. Figure 3.1 illustrates the methods and framework of DRM in which, the links between the stages show the methods that are used to achieve deliverables. Moreover, light demonstrates the process flow. While the bold arrows show the methods and deliverables of each stage.

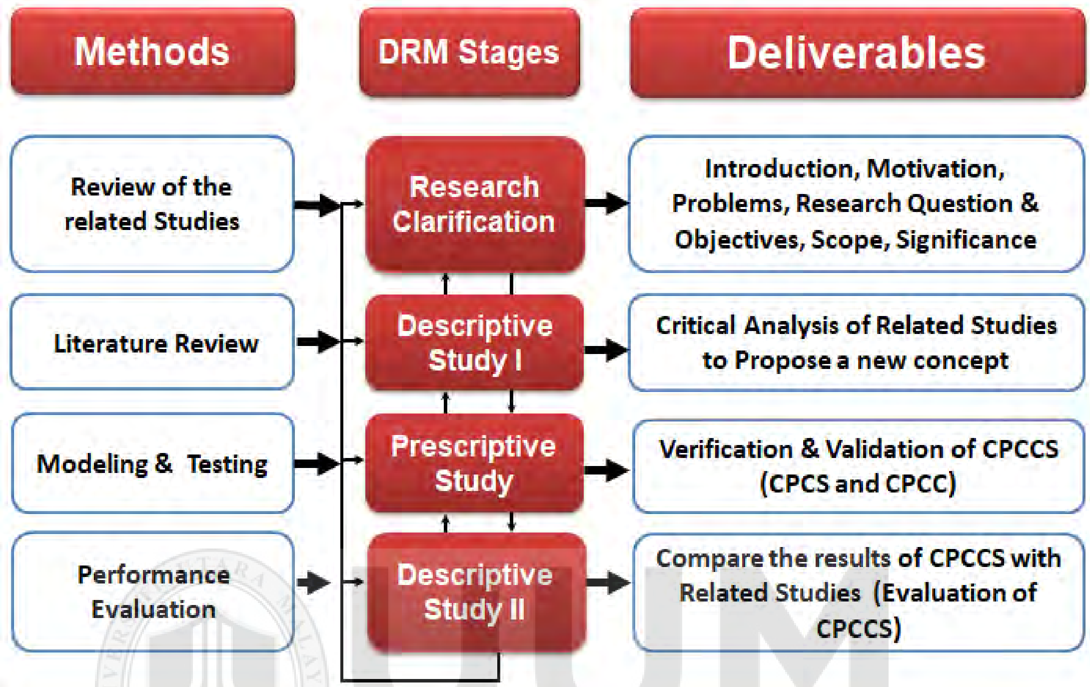

Figure 3.1. Research Approach (Adopted from [133])

3.2 Research Clarification

Research Clarification (RC) provides comprehensive detail to clarify the objectives of the current study. Such as; the aim of this study is to design a cache deployment strategy for NDN to reduce the distance from the subscriber to the available source. Moreover, the proposed strategy decreases the content redundancy and enhances the diversity ratio along the data routing path by the implementation of compound content selection mechanism. In addition, it increases the cache hit ratio through caching the content close to the consumers. Stepwise explanation of Design Research Methodology (DRM) is described by Blessing [133].

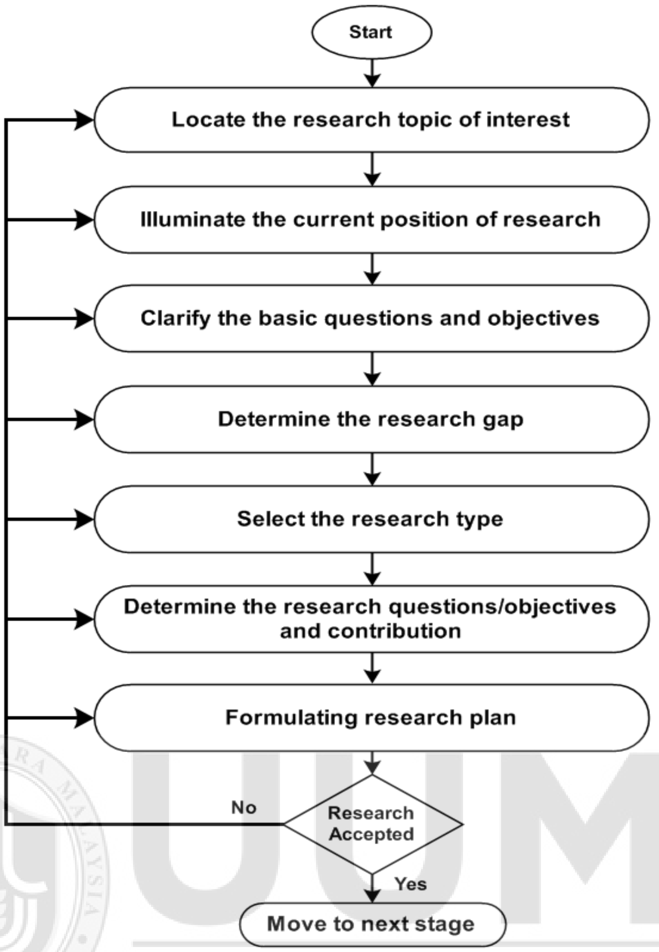

Figure 3.2. Research Clarification (RC) (Adopted from [133])

It is concluded that all the steps of DRM should complement one another in order to achieve better and reliable outcomes for the research problem. Research clarification (RC) is the first phase of DRM that describes the overall research plan in which; the research gap is identified after getting a review of the literature. Moreover, RC is used to clarify the basic objective of this research (develop a cache deployment strategy).

More specifically, this stage focuses on research motivations as well as research questions and objectives for the proposed caching strategy are designed. At the end of this stage, the researchers create a plan in which the scope of the current study is defined. For example, the present research falls within the on-path caching module of NDN. RC covers the whole first chapter for the present study. Therefore, the research will proceed to the next phase if the overall research is acceptable. Figure 3.2 illustrates the overall process of RC phase.

3.3 Descriptive Study-I (DS-I)

As the procedure of RC (expresses the whole research plan) is complete, the research gets transferred to the next stage which is termed as Descriptive Study-I (DS-I). The DS-I is used to understand the present condition of the research in-depth and it implicates a critical literature review of the research area as well as experimental studies. During the development of this research, a comprehensive review in NDN area has recently been deliberated [134] through various experimental studies of NDN existing strategies which critically discussed the issues under discussion to increase the understanding of the previous techniques. DS-I has five sub-stages with many repetitions as explained in Figure 3.3.

The descriptive study is consisting of five stages in which the literature review is explained about the related existing NDN caching strategies. It also defines the focus of the research that is related to the cache deployment strategy. According to the present research, the critical review is conducted of existing caching strategies in DS-I stage. Moreover, the research questions and objectives are refined and the conceptual model is established to resolve the issues derived from the literature.

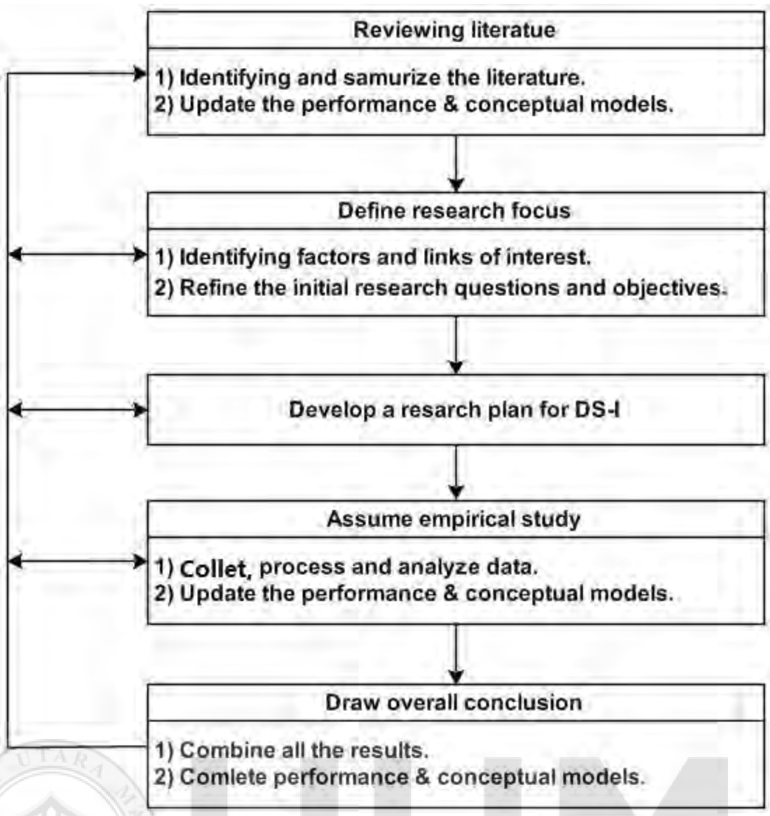

Figure 3.3. Descriptive Study-I (DS-I) (Adopted from [133])

3.3.1 Conceptual Model of Compound Popular Content Caching Strategy (CPCCS)

NDN research approaches need to be verified and validated on a widely used platform (simulation environment). In previous studies [104, 105, 122] it was observed that the ideal structure of the network could affect the overall performance of the network. Cache management is an optimal feature of content centrism, and several researchers have focused on diverse methods of managing disseminated content in networks. Recently, several content caching mechanisms have been developed to increase the efficiency of in-network caching by managing the transmitted content regarding the diverse nature of caching approaches. However, in existing caching mechanisms, several problems related to the multiple replications of heterogeneous content persist increasing memory wastage. To actualize the basic concept of the NDN cache, content caching mechanisms need to implement the optimal objectives and overcome the issues in the data dissemination process [107]. Consequently, in this study, a new flexible mechanism for content caching has been designed to improve the overall caching performance [59].

This section provides a complete description of the proposed content caching mechanism, which is divided into the three phases: In the first phase, the Optimal Popular Content (OPC) is selected by calculating the total number of received Interests for a particular content name. All nodes calculate the number of received Interests for each content using the PIT record and categorize the content into the Optimal Popular Content (OPC) (the content that received most of the consumer Interests) and the Least Popular Content (LPC) (the content that received least consumer Interests) based on the threshold value. The threshold value is the average of the total number of received Interests for all content, and it is changed whenever a new Interest is generated. Therefore, the threshold value shows the average of the total received Interests for all content.

If the number of received Interests for a particular content name is greater than the average of the total number of received Interests for all content, the content is considered to have a high level of interest. Let's assume that the sum of the received Interests for all content is 40, in which 13 and 8 Interests are received for content C1 and Content C5 respectively, and 19 Interests are received for remaining contents.

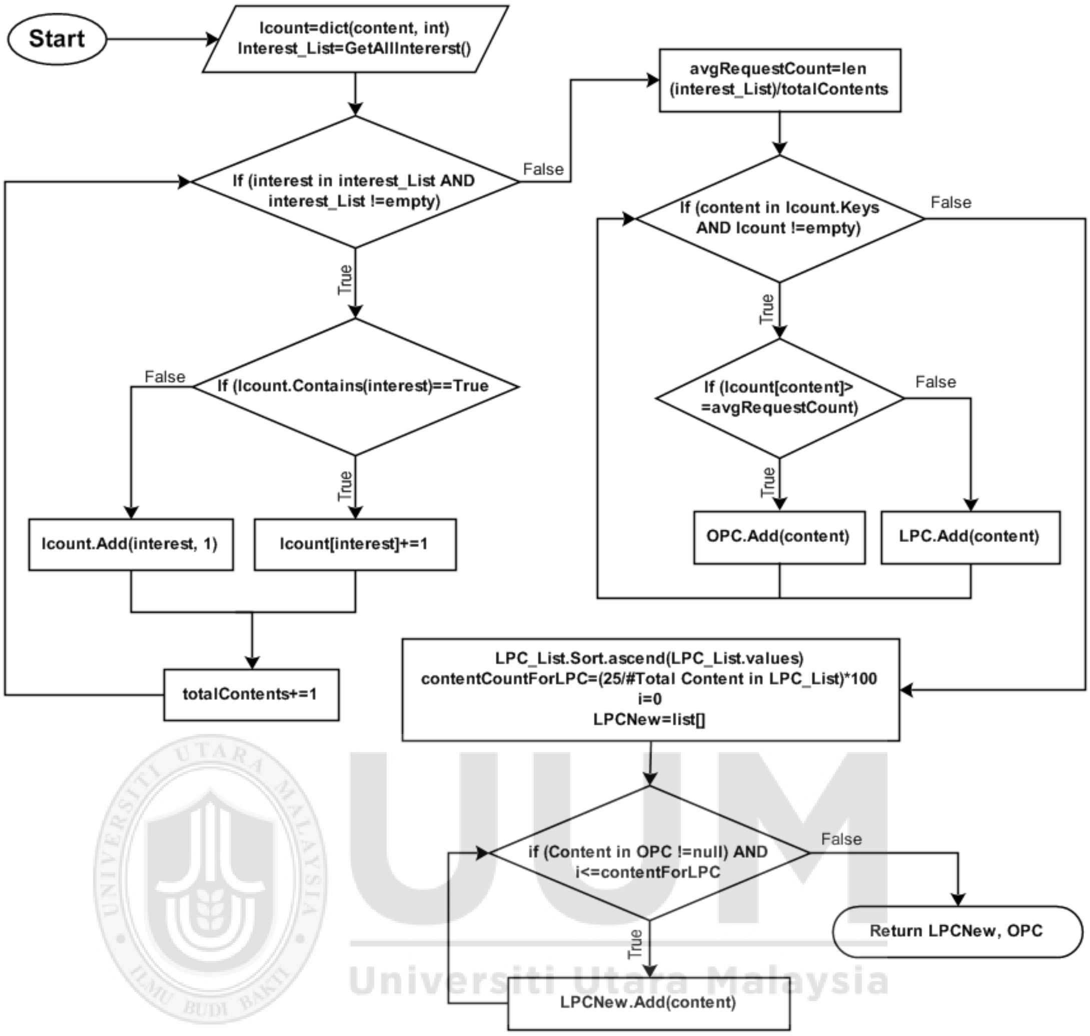

Figure 3.4. Selection of Optimal and Least popular contents

Measure the average of all received Interests, the threshold value is obtained as threshold = 4. Therefore, the content C1 and content C5 suggested as OPC because both contents have received more Interests $(C1 = 13,\ C5 = 8)$ than the threshold $(thershold = 4)$ as shown by Table 2 in Figure 3.5. Meanwhile, if the total number of received Interests for content is less than the average of the total number of received Interests for all content (threshold), then the content considered as LPC.

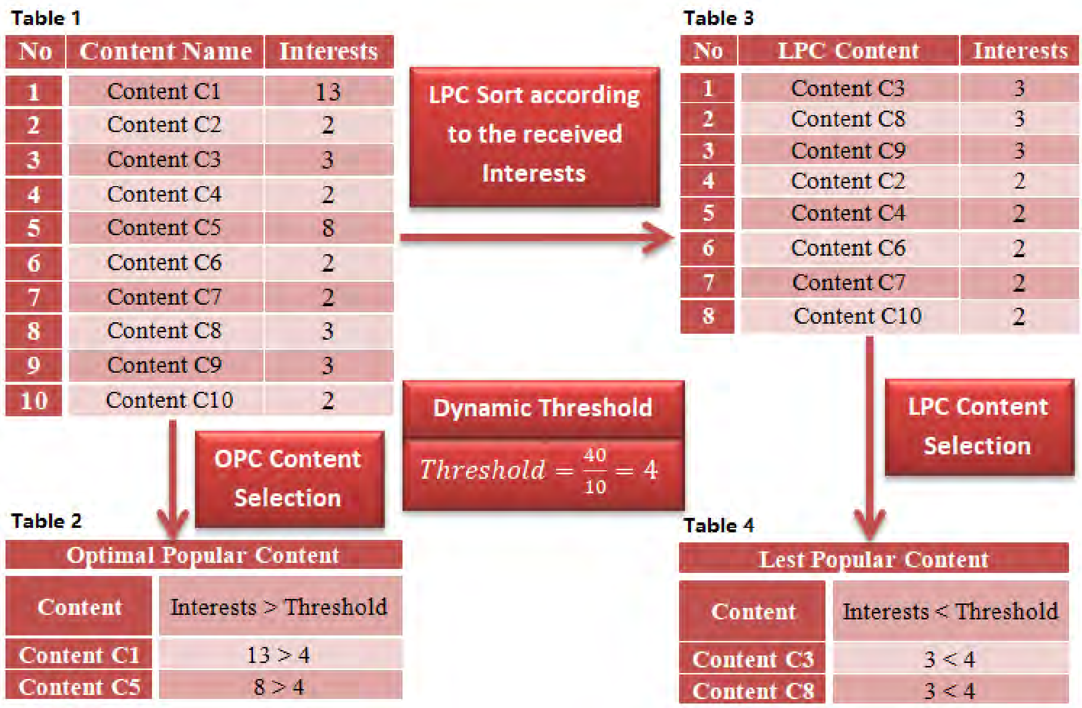

Figure 3.5. Selection of OPC and LPC

According to the content selection algorithm, all the remaining contents are sorted in ascending order regarding their Interest count (illustrated by Table 3 in Figure 3.5) and then one-fourth of the total number of contents are selected as LPC as given by Table 4 in Figure 3.5. Therefore, content C3 and content C8 are selected as LPC because both contents were received more Interests than remaining contents sorted in Table 3. Thus, diverse contents are selected to increase the cache hit ratio. Let's suppose that the total number of contents is 8 in the sorted content list, and then, only 2 contents will be selected as LPC as shown in Figure 3.5. Figure 3.4 (Flowchart Selection of Optimal and Least popular contents) illustrates the content selection mechanism of the OPC and LPC.

The second phase is related to the caching decision about the transmitted content. Information, such as the number of hops, the path distance (from the provider to the consumer), and the location (for caching the transmitted content), are required to cache the content at the intermediate nodes. In this proposed caching mechanism, the number of incoming and outgoing paths is calculated for each node to find the mutual node whenever the content is disseminated from the provider to the consumers. Let's assume that a consumer sends an Interest to the network. However, the cache is empty at the nodes along the path toward the consumer. According to the general NDN practice, the Interest needs to traverse until it matches the required data at the provider's node.

Therefore, the corresponding content is transported through the back path from the provider to the consumer. Furthermore, the copy of the OPC is cached at the all mutually connected nodes of the data routing path before the consumer to reduce the computational overhead and the high cost of the coordination of subsequent Interests and the caching state of the mutual node is shared with its neighbor nodes via a broadcast message to inform them about the location of the OPC. Figure 3.6 illustrates the selection mechanism for mutual node. However, the mutual positions of caching have promised better cache efficiency because most of the Interests are satisfied there.

Although no criteria were defined in earlier caching strategies to handle LPC, the proposed content caching mechanism defines a special criterion for caching to increase content dissemination efficiency and it increases the availability of diverse content close to the consumers.

Figure 3.6. Mutual Node Selection

In the proposed mechanism, LPC is cached only at one mutual node that is placed near the consumers. To avoid the unnecessary usage of the cache, the LPC content caches only at one node because there is less chance to regenerate the Interests for LPC. Hence, most of the cache is used to accommodate the OPC.

In the third phase, to avoid the unnecessary usage of cache storage, all replicas are available for a specific duration; after completion of the content-caching period, they are deleted. In this mechanism, the content life-span is dynamically changed based on the cache storage available. If the cache storage is increased, the life-span will automatically be increased for all cached content. However, if the node has less storage, the content life-span will be decreased.

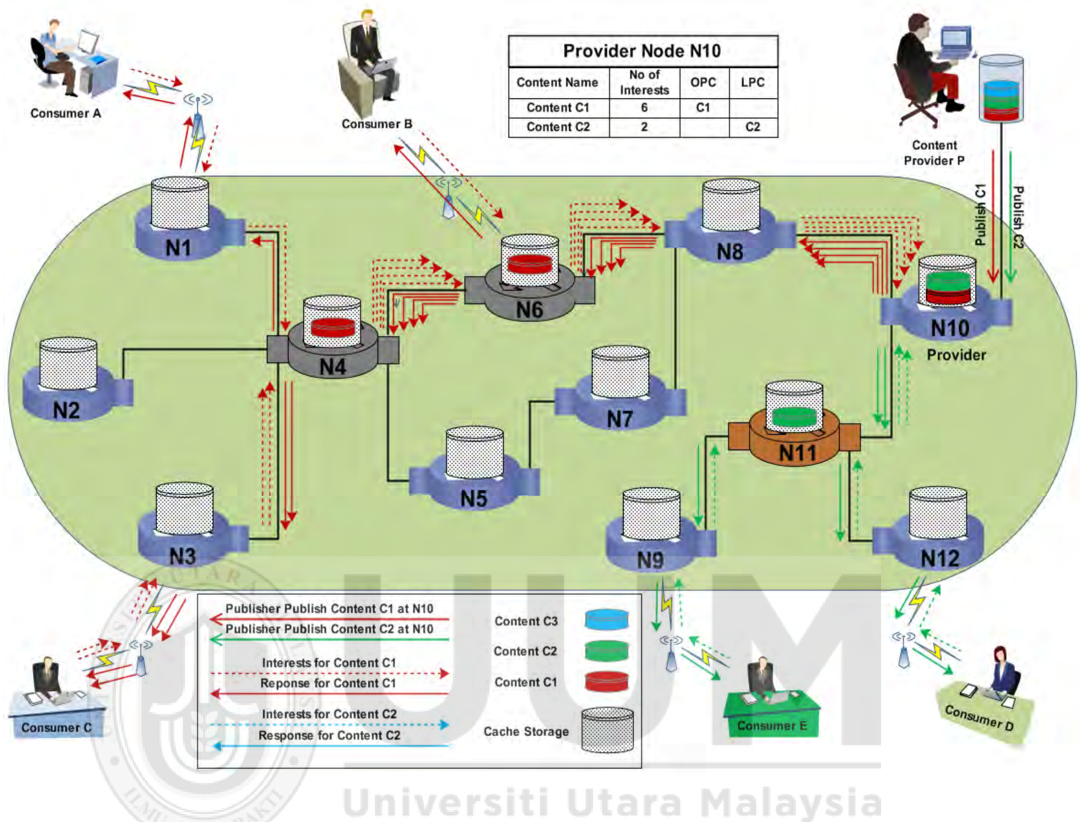

Figure 3.7. Caching mechanism of Optimal and Least popular Contents

To accommodate newly arriving content, CPCCS adopts the Least Recently Used (LRU) content replacement policy to evict old content from the cache after its lifespan expires. For example, if a node has large cache storage, CPCCS would select five seconds as the content life-span, meaning that, the content would be deleted after five seconds of cached time. However, if a node has less available storage, the content life-span would be decreased.

Figure 3.7 illustrates the basic content caching mechanism in CPCCS. In the given scenario, content provider P publishes content C1 and content C2 in a network. The content is initially cached at node N10. After a while, node N10 receives four Interests from consumers A, B, and C. To respond the consumers' Interests, node N10 becomes a provider and sends content C1 to consumers A, B, and C.

In Figure 3.7, the consumer's Interests are indicated by dotted line arrows. The solid-line arrows indicate the responses from the provider. Simultaneously, two Interests for content C2 are received from consumer D and consumer E at node N10. In response, N10 sends content C2 to consumer D and consumer E. According to the proposed caching mechanism, the total number of received Interests for content C1 is 6. Subsequently, content C1 is selected as OPC because it has an extra Interest over the average of the total number of interests received for both content C1 and content C2. Therefore, the caching operations during the transmission of content C1 as OPC are done at mutually connected nodes N4 and N6 because both Node N4 and N6 are mutually connected with interested consumers as A, B and C. Therefore, the subsequent Interests from consumer A, B and C will satisfy from these mutually connected nodes (N4 and N6). Secondly, the content C2 is selected as the LPC content because it has fewer Interests than the average of the total number of received Interests for content C1 and content C2 at provider node N10. Therefore, content C2 will only be cached at one mutual node as N11 that indicating lest distance from the interested consumers D and E as shown in Figure 3.7.

Through this way, the following goals are achieved: the content redundancy and content diversity are improved by increasing the caching of heterogeneous content, the stretch is kept at a minimum, and in turn, the cache hit ratio is increased by caching the content near the consumer.

3.4 Prescriptive Study (PS)

It refers to the design of the proposed CPCCS strategy. With the end goal of this research, systematic demonstration and reproduction process proposed are being employed [134].

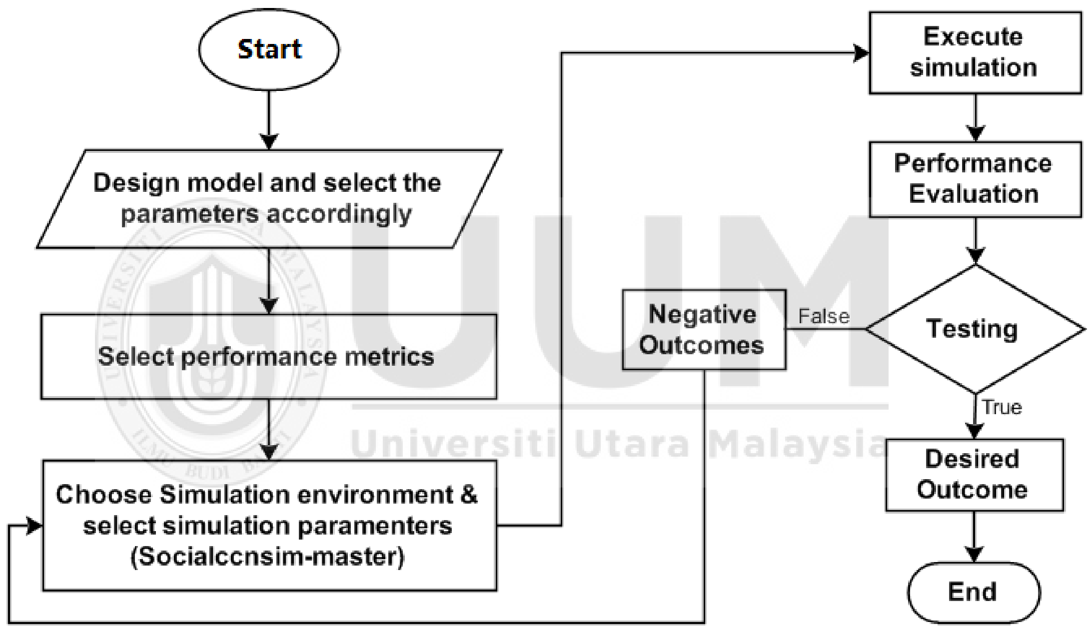

Figure 3.8. Prescriptive Study (PS) (Adopted from [135])

Figure 3.8 demonstrates a square outline depicting the primary studies of the prescriptive stage as indicated by this examination marvel. The first step in this stage is referred to as the proposed strategy (convention) to display. The second and third steps of the given diagram display the improvement which incorporates issues of analysis, objectives, and investigation of the related theories. Besides, it consumes much time as it includes making conventions and presenting the model's complications in a simplified manner. Step 4 and 5 demonstrate model employment and it depends particularly on the decision of the simulation environment. As a final point, detailed acceptance will be covered along with the accompanying subsections. The deliverables of the PS stage are:

- Chapter Four and Five
- Design and implementation of CPCCS mechanisms
- Validation of the CPCCS mechanisms

The PS is the main stage in DRM, as it includes the design of the CPCCS mechanisms. For the purpose of this research, network modeling and simulation process proposed by Guizani *et al*. are being followed [134].

3.4.1 Verification and Validation

It is necessary to verify and validate the proposed study using some tools (simulators) to make it acceptable. Therefore, in this study, the purposed CPCCS needs to be verified. The verification is a conversion of the purposed model to another form satisfactorily to maintain a specific level of accuracy. It may be said that the verification of the proposed model is being used to estimate the accuracy after the execution of the proposed model by using some tools or simulator [136, 137]. First, the model represents the form of flowcharts to make it easily understandable. The flowcharts help to make basic algorithms in the right flow. The verification process will be done through simulation. Simulation is a computer process which is used to calculate the accuracy of the proposed model. It is written in the form of algorithms or pseudo code using any computer language as well as a tool. For example, python language is selected for this study which is requiring SocialCCNSim-Master simulator to simulate the Code of CPCCS. Moreover, it is necessary to be verifying all mechanisms (proposed models) through simulation to make sure that the whole model is coded appropriately and free of errors. The depiction of the proposed mechanism can be verified but the validation is referred to the deployment of proposed caching strategy [138].

The term Verification is related to the static practice to verifying algorithms pseudo-codes. However, the term Validation is related to dynamic mechanism to validate and test the genuine implementation. Validation is the procedure to check as to whether the product meets the requirements and expectations. Validation of the proposed model is defined as "the justification of proposed model inside its specific domain [138].

The model validation can also be explained as the "authentication of the programmed model inside its domain of acceptable range of correctness". Validation gives proof that the model will meet future necessities. Simply, it can be said that the verification is related to the functional correctness (proof of correctness) of the proposed model that intends to check the correctness of the proposed mechanism. The validation is related to the correctness of actual implementation that fulfills all the proposed requirements.

3.5 Descriptive Study-II (DS-II)

The DS-II motivates the evaluation of the designed model and mechanisms. Performance evaluation is the most important part to evaluate any research. In fact, this path provides diverse techniques and platforms to evaluate the proposed work. This will be attained using the third objective. Therefore the performance of the CPCCS Caching strategy will evaluate using the conceptual model of this research to develop a simulation environment and compare its functionality and usability with other NDN-based caching strategies.

3.5.1 Simulation

Simulation procedures are broadly used to represent the actual behaviors of a system. Simulation is a machine-based application that is run on computers using particular software that requires computer programs. By using the simulation setup, it is possible to analyze the performance of the strategies in the controllable, scalable and repeatable manner [139]. For the current research, simulations will perform to evaluate the performance evaluation. From the recent studies, ICARUS [140] is known as a predominant simulator that is only developed for caching of NDN. Conversely, other NDN simulators like SocialCCNsim [102] and ndnSIM [141] are also available to evaluate the performance of NDN caching. Therefore, the base of the current research, a simulation setup is constructed to evaluate the performance evaluation and validation of the CPCCS caching strategy. Thus, SocialCCNSim-Master simulator is selected proposed by Cesar Bernardini [102] to evaluate the performance of the current study.

3.5.1.1 Simulation Setup

Simulation process can be segregated into a number of stages that are used to measure the performance in a disciplined way [142]. These stages are divided into two main categories and these two further divided into several steps as shown in Figure 3.9.

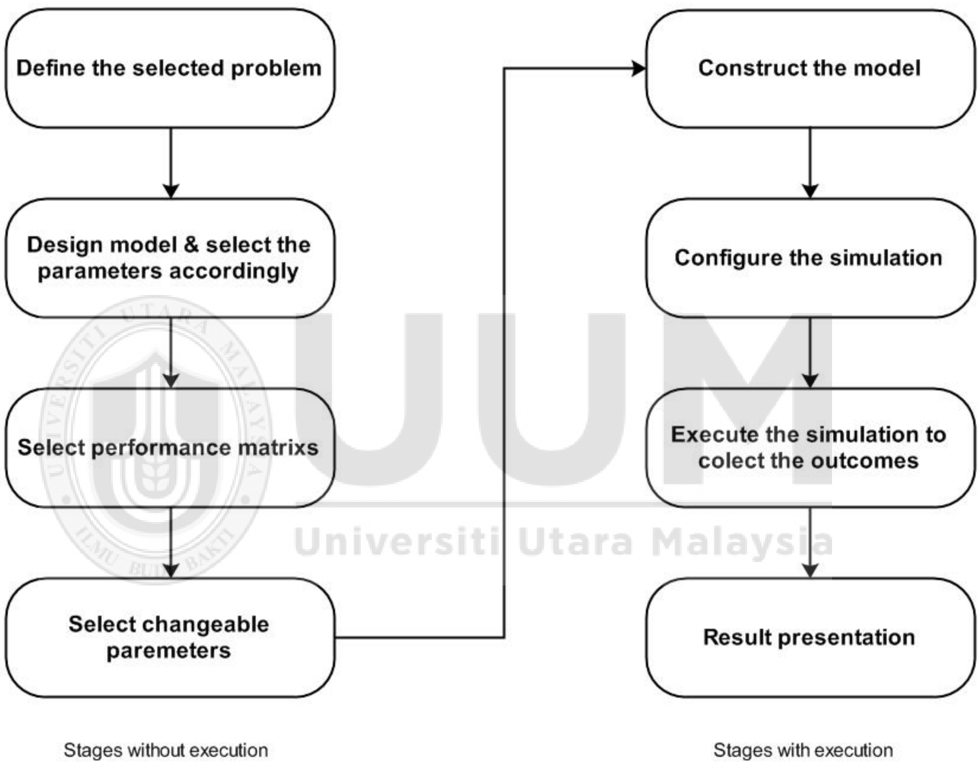

Figure 3.9. Simulation Steps (Adopted from [143])

The given figure is divided into two groups of stages; each group has its four steps defined by Hassan and Jain. These steps illustrate the problem-solving in a well-organized manner, for example, how to select a specific problem, how to make a perfect model as well as the parameters according to the selected design, how to select the appropriate metrics to calculate the performance and how to select the variable parameters. Finally, the model will enter into the simulation environment. In the execution phase, it is explained how to construct models for simulation, how to configure the simulation process, how to simulate the selected model and how to present the results which are received after completing the simulation process.

3.5.1.2 Simulation Scenario

NDN research approaches need to be verified and validated on a widely used platform (simulation environment). In previous studies [104, 105, 122] it was observed that the ideal structure of the network could affect the overall performance of the network. Cache management is an optimal feature of content centrism, and many researchers have focused on the diverse methods of managing disseminated content in networks. Recently, several content caching mechanisms have been developed to increase the efficiency of in-network caching by managing the transmitted content according to the diverse nature of their caching approaches. However, in existing caching mechanisms, several problems related to the multiple replications of homogeneous content persist increasing memory wastage.

To evaluate the proposed content caching mechanism CPCCS, it is necessary to compare its attributes with analogous ones in earlier mechanisms [60, 102]. Therefore, the SocialCCNSim-Master simulator [144] is selected for the present study as the simulation environment for the performance comparison. SocialCCNSim-Master was specially designed to identify the performance of NDN caching and it is widely used to evaluate the several ICN and NDN based caching studies that were published in well-known and high ranked journals such as IEEE communications surveys and tutorials and other [38, 72, 103, 144-148]. It takes data traffic from the Facebook social network topology, which is associated with 4,039 users who each have 44 relationships, and 88,234 friends. SocialCCNSim-Master supports five ISP-level topologies (i.e., GEANT, Abilene, DTelekom, Tree, and Tiger). For the present research GEANT, DTelekom, Tiger, and Abilene, topologies are selected. The reason is that GRANT was developed especially for research, education, and innovation communities across the globe. It is pan European network that is used to interconnect Europe's National and Education Networks (NRENs). GEANT has the ability to connect more than 50 million consumers at 10,000 organizations. Moreover, GEANT provides high bandwidth and secure high capacity (50,000km) network with an increasing range of services that allow the researchers to cooperate with one another wherever they are located. GEANT has 22 nodes contented to each other. The objective of the GEANT topology is to transfer large amounts of data across the nodes. Its structure is perfect for testing and comparing our simulation and observing the efficacy of CPCCS.

On the other hand, Abilene formed a high-speed backbone by deploying cutting edge technologies not yet generally available on the scale of a national network backbone. Abilene was a private network used for education and research but was not entirely isolated, since its members usually provide alternative access to many of their resources through the public Internet. Sometimes it referred to Intetnet2 network topology. Abilene topology was designed with the sole object of transferring large size data across nodes. Its building was ideal to test and compare our simulation parameters to record the cache-hits and observe the efficacy of our cache managements. Abilene heterogeneous nature depicts eleven (11) station sites as nodes. Its hierarchical structure provides the flexible neighbor composition needed to test NDN cache deployment strategies. GEANT and Tiger topologies have 22 nodes and DTelekom consist of 68 nodes contented to each other. This forms the basis of our topology selection to test the performance of the strategy for the heterogeneous Internet. To avoid unnecessary cache usage, the Least Replacement Policy (LRU) is used to replace the old content with the new. LRU is considered the most efficient content replacement policy because of its flexible structure and high performance.

3.5.2 Performance Metrics

How each node makes a caching decision is a major problem in on-path caching that impacts the performance of the content delivery process. For example, popular content needs to be cached at the node where it will be demanded next [149]. The information content can be cached opportunistically at the delivery path for the consequent Interests [64]. The efficiency of caching is dependent on different types of performance metrics that are necessary to improve the content caching strategy by considering the cache size, which is small compared to the communicating data content. Therefore, some essential performance metrics are included in this study to critically compare NDN caching strategies and determined which show the optimal performance. Metrics were used to test the effectiveness of cache management strategies. For the present study four well-known metrics content diversity, cache hit ratio, redundancy, and stretch were selected to compare the performance of the proposed caching strategy. Some of the performance analysis of NDN caching is shown in Figure 3.10.

3.5.2.1 Content Diversity

This defines the number and rate of caching distinct content in the heterogeneous network or the ratio of how different sets of content accumulate in the network cache. The resulting rate is computed to mitigate high information replication [106]. Diversity defines the performance of cache replication. It provides and computes the ratio of distinctly stored unique sets of content in the cache-able devices.

3.5.2.2 Content Redundancy

Content redundancy is a significant metric to assess homogeneous data dissemination in CCN. In CCN, redundancy is the number of content copies cached at more than one location in the network with fewer referrals or less unsubscribing [150]. Redundancy is therefore presented for each network setup and accordingly during Interest and content dissemination.

3.5.2.3 Cache Hit Ratio

The cache hit ratio is the portion of content Interests satisfied by caches that are implemented within the network. Several studies in the database and networking domain have emphasized the performance of the network by improving the cache-hit ratio. This metric shows the potential of the caching strategy to moderate the amount of redundant content.

3.5.2.4 Stretch Ratio

Stretch refers to the number of hops (routers) that need to be covered by a consumer's Interest between the content provider (where the hit occurs) and the consumer. Whenever the consumer sends an Interest for some content, the Interest needs to be covered by a number of hops to reach the source for the required content. While the Interest can find a copy of the required content from any of the routers that appear on the path, sometimes it has to go to the main content provider server, which increases the average cost. Therefore, each caching strategy tries to reduce the maximum number of hops between providers and consumers.

3.6 Research Steps

With the purpose to fulfill the objectives of this research following steps are needed to be followed:

1. Conduct a survey of existing probabilistic and popularity-based caching mechanisms termed as cache deployment and content placement strategies to identify the critical issues.
2. Conduct a critical analysis of popularity-based caching strategies to find the difficulties due to the static threshold.
3. Analyze the popularity-based content placement strategy to find the ideal positions for the caching of least and optimal popular contents.
4. Design a content selection mechanism according to the Interests frequency by enhancing the dynamic threshold and differentiate the optimal and least popular contents.
5. Design a position selection mechanism in order to optimize the content store of a network node for least and optimal popular contents.

6. The proposed strategy is extensively and comparatively studied with other popularity-based caching strategies in a simulation environment to investigate the strength and weaknesses of the proposed caching strategy.

3.7 Summary

This chapter explained in detail the research approach to make sure that the research goals are materialized. This part of the research deliberates on developing a content deployment strategy for NDN cache management. In this chapter, the whole research is divided into four main parts that were outlined with DRM. The first part is the Research Clarification (RC) phase that is used to explain the initial idea of the current research. The purpose of using RC is to recognize and improve a research problem, research questions, and research objectives. Therefore, both objectives and questions practically and academically have to be meaningful and genuine. The second part of this research is Descriptive Study-I (DS-I) phase. It deliberates the basic steps to achieve an appropriate understanding of the present condition as it helps to design a reference model by recommending the conceptual model. In this chapter, the third part Prescriptive Study (PS) has been used to examine the already existing methods in designing the proposed strategy. The last part is a DS-II which emphases the evaluation of the proposed strategy and it is purely related to achieving the third objective. The implementation of this proposed research will be explained in the next chapters.

CHAPTER FOUR
COMPOUND POPULAR CONTENT CACHING STRATEGY

4.1 Introduction

NDN research approaches need to be verified and validated on a widely used platform (e.g., simulation environment). In previous studies [104, 105, 122] it was observed that the ideal structure of the network could affect the overall performance of the network. Cache management is an optimal feature of content centrism, and many researchers have focused on the diverse methods of managing disseminated content in networks. Recently, several content caching mechanisms have been developed to increase the efficiency of in-network caching by managing the transmitted content according to the diverse nature of caching approaches. However, in existing caching mechanisms, several problems related to the multiple replications of heterogeneous content persist increasing memory wastage. Therefore, to actualize the basic concept of the NDN cache, content caching mechanisms must implement the optimal NDN objectives and overcome the issues in the data dissemination process faced by the recent mechanisms [107]. Consequently, in this study, a new flexible mechanism for content caching has been designed to improve the overall caching performance.

In this chapter, a proposed content selection mechanism pronounced as Compound Popular Content Selection (CPCS) mechanism is designed to increase the number of diverse contents along the data delivery path and reduce the number of redundant content replications. In addition, the CPCCS strategy is extensively and comparatively studied with other NDN-based caching strategies such as, Hop-based

Probabilistic Caching (HPC), Most Popular Cache (MPC), LeafPopDown, MAx-Gain In-network Caching (MAGIC), Cache Capacity Aware Caching (CCAC), and Popularity-Driven Coordinating Caching (PDCC), through simulations.

4.2 Content Placement Concept

Cache management is very important because on-path caching cannot generate efficient results. Cache-less networks can produce efficient results. The management of the cache depends on how the cache is deployed and managed by the network. The effective management of the cache demonstrates a mechanism that places a copy of the transmitted content into the network caching system efficiently [14]. This mechanism reduces the server load and latency of subsequent requests. Further, it increases the cache hit rate and availability of data [15]. Therefore, the intention of on-path caching is to reduce the usage of the resources of the Internet. For this purpose, on-path caching delivers several cache management strategies that help in overcoming the data transmission problems such as heterogeneous content placement, cache hit ratio, content redundancy, and path length between the publisher and the consumer. The management of the cache depends on how the cache is deployed and managed by the network. The effective management of cache demonstrates a mechanism that places a copy of the transmitted content into the network caching system efficiently [14]. Therefore, the intention of on-path caching is to reduce the usage of the resources of the Internet [15].

For the efficient caching distribution, several popularity-based content placement strategies were proposed. Thus, a number of diverse types of cache management strategies such as; MAx-Gain In-network Caching (MAGIC) [73], WAVE

Popularity-based Caching Strategy, Hop-based Probabilistic Caching (HPC), LeafPopDown, Most Popular Cache (MPC), and Cache Capacity Aware Caching (CCAC) are explained in previous chapters.

To define optimal popularity-based caching strategy, it needs to find the content's popularity by calculating the total received Interests for a particular content then find the location for caching the popular content. After caching of popular content an age (time span) is associates to each content after which the content will be deleted automatically for the accommodation of new coming contents. Moreover, it is necessary to consider the available cache storage and cost in terms of distance from the consumers to the node selected for caching to fulfill the subsequent requirements.

Usually, the Graph is used shows the relationship between objects [151]. In this context, the graphs are made up of nodes, vertices or points which are connected by arcs, lines, and links that cite as edges [152].

Generally, graphs are divided into an undirected and directed graph. In undirected graphs, the edges or links have no directions between two vertices and the directed graphs have directions between two vertices [153]. In networking, the graph is broadly used to construct to understand the network structure [154]. The graph is largely used in NDN-based applications because it is most appropriate to express the NDN-based caching architectures [155]. In the graph, the connections among the midpoints edges and nodes are found to show the caching of NDN. An undirected graph $G(V, E)$ shows the network connectivity in which $V = \{v_1, v_2 ...., v_n\}$ is the set of nodes and $E = \{e_1, e_2, ..., e_n\}$ shows the set of edges as communication paths among the network nodes.

Suppose, there exist NDN-based graph and a large number of consumers are connected with several nodes in a network [155]. Some consumers are hierarchically connected and some nodes are inherent. To describe the basic concept of NDN Caching, assume that NDN environment represented in the form of undirected graph $G(V, E)$ in which $V = \{v_i | i \in I_v\}$ is the set of vertices and $E = \{e_i | Iv \in I_e \Lambda\, e_i \subseteq V \Lambda\, e_i \neq 0\}$ are respective connectors that connects the different network nodes and I_v and I_e are the index sets of vertices and edges.

Assume that all the contents having uniform size as D = {d} where d is the smallest chunk of content in the network. In order to cache content during its transmission from provider to consumer, it is essential to have the information about the content's availability and its popularity with respect to time because either the popularity is increasing or decreasing as the changing in time. Moreover, the location also affects the content's popularity because many consumers can download content if it is cached at the central position rather than content cached at edge position. However, the content availability, popularity and the consumer's demands for cached contents can be achieved using the past data dissemination information of a network. Therefore, a network node can calculate the content availability and popularity from the past information of cached content.

Therefore, the availability of a content x ∈ X cached at a network node v ∈ V can be achieved by the following equation 4.1:

$$A_v^x(t) = (0,1) \tag{4.1}$$

where A shows the availability of content x cached at v and t is the time span during the content x is available at node v. On the other hand, the availability of content will be zero if content x is not cached at node v with respect to time t given by the equation 4.1.

According to the caching requirements for a specific content x, we select C_x which shows the number of chunks cached at a network node because multiple chunks related to different content can be found at a node. However, the content size can be defined as $S_x = C_x.x$ and therefore we can calculate the total size of all content cached at node v will be represented in equation 4.2:

$$S_v = A_v^x S_x \tag{4.2}$$

The equation 4.2 shows that the amount of all content sizes possibly cached at node v. Therefore, the transmitted contents can be cached at intermediate nodes to perform subsequent content's transmissions as the new request for cached content is arrived.

4.2.1 Content's Popularity

The popularity of contents in a network can be defined through probability matrix and it can be calculated for the current scenario as in given in equation 4.3:

$$P_x(t) = \frac{\delta_v^x(t)}{\Upsilon_v^x} \tag{4.3}$$

The equation shows the probability of consumer Interests at a for content x with respect to time t. $\boldsymbol{\delta_v^x(t)}$ represents the number of received Interests at node v for content x in time t and $\tilde{\Upsilon}_v^x$ shows the total number of received Interests for all the contents at node v. It is important to consider that the content's popularity is used to calculate by taking the sum of all the Interests for individual content or it can be calculated through the frequency of Interests for a particular content name.

4.2.2 Selection of Popular Content

For efficient content caching, it is necessary to find the relation between received Interests and cached content. Content is considered as popular if the frequency of Interests increases for a particular content because the nodes continuously receive consumer's Interests for cached content. Therefore, it is necessary to find the relationship between cached content and the node because all the nodes have different impacts on received Interests and content delivery according to their locations in the network. Therefore, the Interest frequency can be calculated by using the following equation 4.4:

$$F_v^x(t) = \frac{r_v^x(t)}{R_v^X} \tag{4.4}$$

Where the $r_v^x(t)$ represents the total number of Interests responses from node v by sending content x to the desired consumers in specific time span t and R_v^X shows the total number of responses to the overall received Interests for all the contents cached at node v.

4.2.3 Interest's Validity

We assume that each cached content associated with a specific time span after which it will delete from the cache to accommodate the new content. Therefore, we consider this time as content's validity in which the number of consumer's Interests are satisfied thus it is called Time of Validity (ToV). The ToV is different for all the content and it varies based on the received Interests. If the content is continuously receiving Interests, the content's validity will increase. It can be defined as for each received Interest the content's validity is increased at node v and hence the average lifespan for particular content is specified by the consumers' Interests. Therefore, it makes the node enable to take the decision whether the cached content is still requested by the consumers or not.

On the other hand, if a content is not demanded by the consumers, it needed to be evicted to make room for the accommodation of new contents. Thus, the contents' validity helps to identify the usage of cached contents and it is significant to avoid the unnecessary usage of the in-network cache. To improve the quality of caching services the content replacement policies is used for the eviction of unnecessary cached contents. Consider $t_v^{x_f}$ is the time in which the consumers' Interests for content x are satisfied from network node v and the x_f represents final Interest for content x is satisfied. Thus, the average validity time span for content x can be represented as $t_v^{x_{av}}$.

4.2.4 Content's Validity

Like Interest's validity, the content's validity is considered as most significant for the improvement of NDN caching performance:

$$\Psi_v^x(t) = e_v^{-\delta t_v^{x_{av}}} \tag{4.5}$$

$\delta \in [0, 1]$ is the adjustable parameter that is used to adjust the exponential decay. The validity for all the contents is different because their arriving time can be different or some contents are most popular and receiving the consumer's Interests continuously rather than those have been downloaded for a short time span.

Let's assume t_c shows the current time interval that can be less or rater than the time interval at which the content x was last responded $t_v^{x_f}$ and the average Interest validity time span. According to this decay, a network node can decide for how long content can be cached at a node and the value of this decay is different for divers' replacement policies. Therefore, the value of decay will be change according to the selected replacement policy. The reason is that, if a content receiving consumers' Interests continuously, the content's validity will be increased because the validity will be reinitiated for those contents that are selected as most interested again. On the other hand, if the average Interests validity is expired, the content will be evicted from the node because the content's validity follows the exponential decay as the consumers' Interests decreases for cached content.

4.2.5 Eligibility

The caching mechanism considered content's availability and its popularity for the selection of most interested content to be cached close to the consumers. Therefore, it is significant to determine the relationship between contents' availability and its popularity. As it is already mentioned in the above equations, the contents' validity is Ψ and its popularity is P_x while A_v^x shows the availability of content x at node v and F_v^x is the frequency of Interests satisfied by the content x at node v. Therefore, the availability of content x at node v and the popularity of content x can be calculated using equation 4.6:

$$Ц_v^x = \alpha.\Psi_v^x P_x + (1 - \alpha)\, A_v^x F_v^x \tag{4.6}$$

Where α parameter shows the popular content x is available at node v or not. The first part of the equation shows the popular content is cached or not at node v while the second part represents the content is already available at node v.

The equation considered both popularity and availability to find the eligibility of content x. Ψ_v^x shows the validity of content x at node v while P_x represents consumers' Interests probability (popularity) for content x and F_v^x shows the frequency of received consumers' Interests for content x at node v. Therefore, equation 4 shows the contents' availability is proportional to the consumers' Interests for a specific content x. Thus, the $Ц_v^x$ shows the relation that allows a network node v to take the decision for the caching of content x while considering its availability and popularity according to the receiving consumers' Interests frequency. Moreover, the equation considered content popularity as well as content recency. In order to avoid the unnecessary computation for content eligibility, (content x is not cached at node v) the second part of the equation is assigned to zero.

4.2.6 Usage of cache node

The usage of a network node depends on its position. Usually, the NDN network node is associated with cache storage that is used to provide memory for content caching. Therefore, the corresponding cache storage available within each node is represented by 4.7:

$$S_v^t = S_v^u + S_v^f \tag{4.7}$$

where S_v^t shows the total cache storage, S_v^u represents the used cache storage by the cached contents and S_v^f shows the free cache storage at node v. It is important that the most popular content consumes cache storage for a long time due to a large number of received Interests and therefore, these contents yields extra cost because such contents are cached to satisfy the potential consumer's Interests. Thus we can calculate the cost at a node v by using the following equation 4.8:

$$f_v^C = \frac{1}{X} \Sigma \text{Ц}_v^x \rho_x + \text{Ḉ} \tag{4.8}$$

where $Ц_v^x$ shows the content profile in which availability and popularity is measured, $\boldsymbol{\rho_x}$ represents the cache storage in percentage used by content x and $\acute{Ç}$ is used to represents the additional cost in terms of mobility, power consumption.

4.3 Proposed Caching Strategy

In previous studies, it was observed that the ideal structure of the network could affect the overall performance of the network. Cache management has been seen as an optimal feature of content centrism because a large number of researchers have focused on the diverse forms of managing disseminated contents in networks. Recently, several content caching mechanisms have been developed to increase the efficiency of in-network cache by managing the transmitted content according to the diverse nature of caching approaches. However, in recent caching mechanisms, several problems persist, which are related to multiple replications of heterogeneous content, which has increased memory wastage. Therefore, to actualize the basic concept of the NDN cache, content caching mechanisms are required to implement the optimal objectives of NDN to overcome the issues in the current data dissemination process by recent mechanisms [107]. Consequently, in this study, a new flexible mechanism for content caching is designed to improve the overall caching performance.

4.3.1 Compound Popular Content Caching Strategy (CPCCS)

The basic reason behind the efficient cache management is the limited available capacity of cache storage because the amount of transmitting contents is massive as compared to the available cache size. Therefore, the maximum cache efficiency can be achieved by implementing a suitable cache management strategy. However, there is a big gap still exist to find an efficient cache management strategy that should cache content close to the consumers in such a way to increase overall caching performance. Hence, the management of in-network cache is required to find a way to known pre-location of caching where the content will be retrieved subsequently to increase the availability of most interested contents. In order to overcome the difficulties of cache management, we propose a new caching strategy to improve caching performance using the contents' utility.

Caching strategies were developed regarding their different functionalities to accommodate the disseminating contents in a different way to increase the efficiency of the overall caching services. The intention of the proposed study is to increase the availability of the most popular and least popular contents as well by decreasing the path length in terms of the stretch between the consumer and provider. Several popularity-based caching strategies have recently been proposed but all these strategies only focus on the high popularity of contents. Therefore, the reaming unpopular and least popular contents are avoided for the caching services and there are no criteria developed for the caching of these contents. However, the proposed caching strategy is defined as new criteria in which the content's popularity is calculated by taking the average of received Interests.

To cope with the present strategy we define a relationship between the availability and content's popularity in a network. Therefore, the caching decision is accomplished after finding the utility of content and to accommodate the disseminated content (based on their popularity) alongside the data delivery path and maximizes the availability of diverse contents.

Algorithm 1. Selection of OPC and LPC contents

Input:

  interest_List: List which contains interest requests

Output:

  OPC_List: Optimal popular content list

  LPC_List: Least popular content list

```
Procedure GetPopularContent (interest_List):
 Interest_request_List = [ ]
  for content in interest_List do:
    if content ∈ Interest_request_List then
        increment count(content)
     else:
        Interest_request_List.Add(content.New )
        Set count(content.New) = 1
      End
     increment count (total_Contents)
   End
//Loop Ends
  Calculate AVG (request_Count)
       AVG (request_Count) = #distinct_interests_requests / #all_requests
  for content_req ∈ Interest_request_List do
     if content.count > avg_RequestCount then
        OPC_List.Add (content)
     else:
        LPC_List.Add (content)
     End
```

```
    End
return OPC_List, LPC_List
```

This section provides a complete description of the proposed content caching mechanism, which is divided into the three phases: In the first phase, the Optimal Popular Content (OPC) is selected by calculating the total number of received Interests for a particular content name. All nodes calculate the number of received Interests for each content using the PIT record and categorize the content into the Optimal Popular Content (the content that has received most of the consumer Interests) and the Least Popular Content (LPC) (the content that has received the least number of consumer Interests) based on the threshold value.

The threshold value is the average of the total number of received Interests for all content, and it is changed whenever a new Interest is generated. Therefore, the threshold value shows the maximum average of the total received Interests for all content. If the number of received Interests for a particular content name is greater than the average of the total number of received Interests for all content, the content is considered to have a high level of interest. For example, the sum of the received Interests for all content is 30, in which 20 Interests are generated for content C1, and 10 Interests are received for content C2. Taking the average of all received Interests, the threshold value is 15. In this case, content C1 is the OPC because it received more Interests (20) than the threshold (15). Meanwhile, if the total number of received Interests for a content item is less than the average of the total number of received Interests for all content, then the content considered as LPC.

Let's assume that the sum of the received Interests for all content is 40, in which 13 and 8 Interests are received for content C1 and Content C5 respectively, and 19 Interests are received for remaining contents. Measure the average of all received Interests, the threshold value is obtained as threshold = 4. Therefore, the content C1 and content C5 suggested as OPC because both contents have received more Interests ($C1 = 13,\ C5 = 8$) than the threshold (thershold = 4) as shown by Table 2 in Figure 3.5. Meanwhile, if the total number of received Interests for content is less than the average of the total number of received Interests for all content (threshold), then the content considered as LPC.

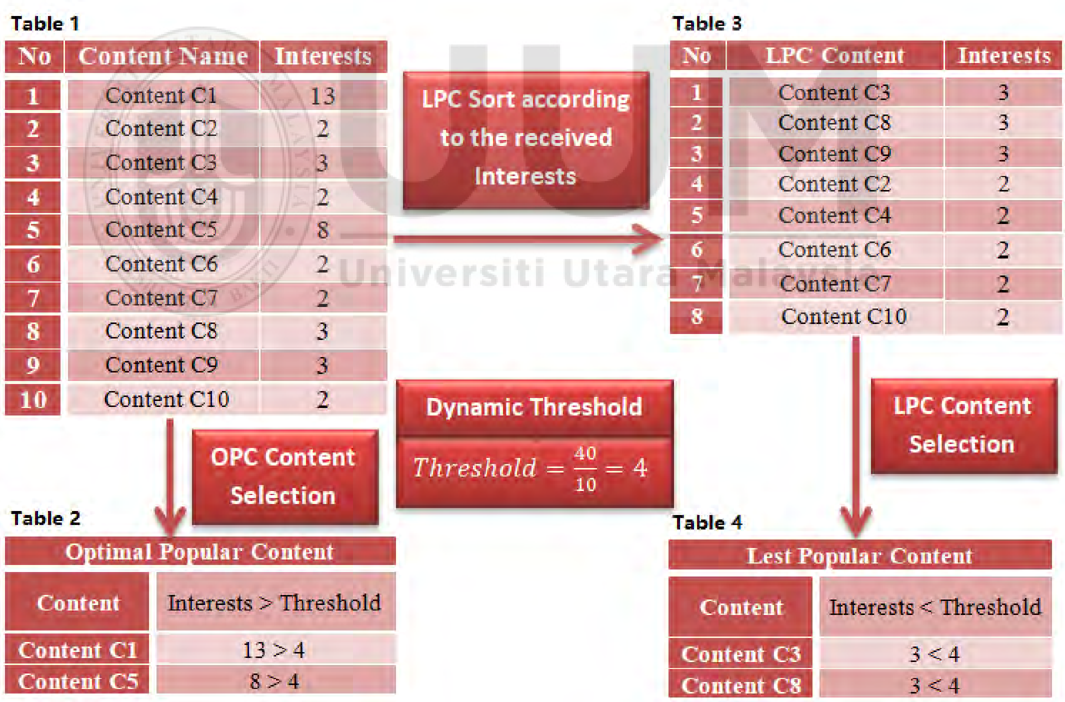

Figure 4.1. Selection of OPC and LPC

According to the content selection algorithm, all the remaining contents are sorted in ascending order regarding their Interest count (illustrated by Table 3 in Figure 4.1) and then one-fourth of the total number of contents are selected as LPC as given by Table 4 in Figure 4.1. Therefore, content C3 and content C8 are selected as LPC because both contents were received more Interests than remaining contents sorted in Table 3. Thus, diverse contents are selected to increase the cache hit ratio. Let's suppose that the total number of contents is 8 in the sorted content list, and then, only 2 contents (Content C3 and Content C8) will be selected as LPC as shown in Figure 4.1. Algorithm 1 (Selection of Optimal and Least popular contents) illustrates the content selection mechanism of the OPC and LPC in CPCCS.

The second phase is related to the caching decision about the transmitted content. Information, such as the number of hops, the path distance (from the provider to the consumer), and the location (for caching the transmitted content), are required to cache the content at the intermediate nodes. In this proposed caching mechanism, the number of incoming and outgoing paths is calculated for each node to find the mutual node whenever the content is disseminated from the provider to the consumers. Let's assume that a consumer sends an Interest to the network. However, the cache is empty at the nodes along the path toward the consumer. According to the general NDN practice, the Interest needs to traverse until it matches the required data at the provider's node.

Therefore, the corresponding content is transported through the back path from the provider to the consumer. Furthermore, the copy of the OPC is cached at the all mutually connected nodes of the data routing path before the consumer to reduce the computational overhead and the high cost of the coordination of subsequent Interests and the caching state of the mutual node is shared with its neighbor nodes via a broadcast message to inform them about the location of the OPC. Algorithm 2 illustrates the selection mechanism for mutual node.

However, the mutual positions of caching have promised better cache efficiency because most of the Interests are satisfied there. Although no criteria were defined in earlier caching strategies to handle LPC, the proposed content caching mechanism defines a special criterion for caching to increase content dissemination efficiency and it increases the availability of diverse content close to the consumers. In the proposed mechanism, LPC is cached only at one mutual node that is placed near the consumers. To avoid the unnecessary usage of the cache, the LPC content caches only at one node because there is less chance to regenerate the Interests for LPC. Hence, most of the cache is used to accommodate the OPC.

Algorithm 2: Selection of mutual node

Input:

Requested Content Node: Node where content found

Requested Content: Content which is needed by user

Nodes: all nodes in a path

InterestList: List in which all interest requests are saved

Output:

OPC and LPC Content cached

Procedure *Mutual* *(RequestedContentNode, RequestedContent, InterestNodes, allNodes, interestList)*

```
RequestedContentNode={ } #Node on which the required  content exist
RequestedContent ={ } #content for  which consumer sent the request
InterestNodes=[ ]    #Interest (shows consumer's request)
```

```
5  Nodes_List=[ ]
6  for Node N ∈ Nodes_List do:
7      for content c ∈ N do
8          if c == RequestedContent then
9              set RequestedContentNode=N
           End
       End
   End
10 pathsToContent_List = [ ]
11 for each Node N ∈ InterestNodes do
12     path=getPath(N to RequestedContentNode)
13     pathsToContent_List.Add(path)
   End
```

In line 6 to 9 of pseudo code the algorithm finds the specific Node (Router) on which the required content is reside and it is denoted as **RequestedContentNode**. From line 10 to 13 the algorithm finds paths to the **RequestedContentNode** from each other node of the network using shortest path algorithm. # Node is used to stores the nearest mutual intermediate node between multiple paths from the requested Node (where the interests are generated) to the content Node (where the requested content is found). Node stores as keys and the distance as values

```
14 mutualNodes = GetMutualNodes(pathsToContent_List)
15 FindAndSetContentWithShortPath(centralityNodes)
```

Procedure: GetMutualNodes(pathToContentList)

```
17 for node in pathToContentList.paths do
18     if node.ConnectedUsers.Count()>1 then
20         mutualNodes.Add(node)
       End
   End
return mutualNodes
```

```
21 Procedure FindAndSetContentWithShortPath (centralityNodes)
22    Node n;
23 for Node N in mutualNodes do
24     for Node M in mutualNodes do
          if N.Length<=.M.Length then
             n=N
           End
        End
25  n.cache.contentPlacement(OPC)
26  n.cache.contentPlacement(LPC)
```

At line 14 the algorithm calls a functions ***GetMutualNodes*** which finds the mutual nodes on which maximum paths (paths obtained earlier in code) meets. From line 23 to 24 the algorithm finds the node ***n*** which have shortest path to ***RequestedContentNode***. At line 25 and 26 the algorithm store the LPC and OPC contents on the node n which algorithm finds earlier at line 24.

In the third phase, to avoid the unnecessary usage of cache storage, all replicas are available for a specific duration; after completion of the content-caching period, they are deleted. In this mechanism, the content life-span is dynamically changed based on the cache storage available. If the cache storage is increased, the life-span will automatically be increased for all cached content. However, if the node has less storage, the content life-span will be decreased. To accommodate newly arriving content, CPCCS adopts the Least Recently Used (LRU) content replacement policy to evict old content from the cache after its lifespan expires. For example, if a node has large cache storage, CPCCS would select five seconds as the content life-span, meaning that, the content would be deleted after five seconds of cached time. However, if a node has less available storage, the content life-span would be decreased.

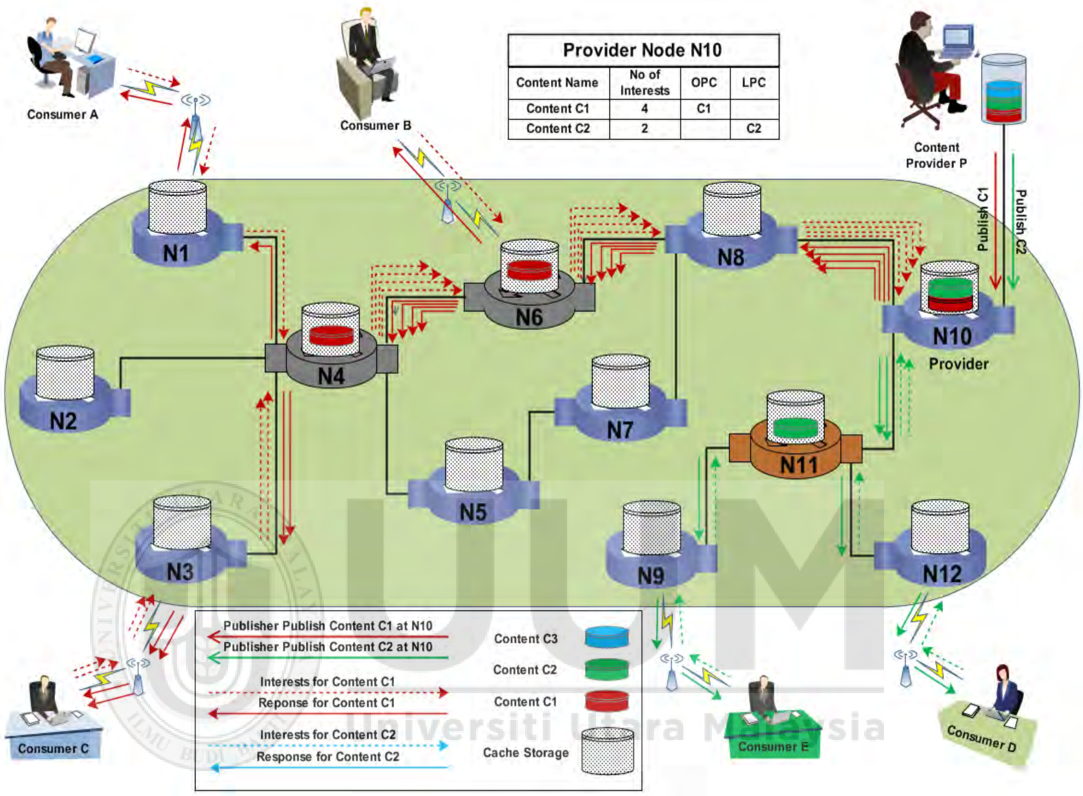

Figure 4.2. Caching mechanism of Optimal and Least popular Contents

Figure 4.2 illustrates the basic content caching mechanism in CPCCS. In the given scenario, content provider P publishes content C1 and content C2 in a network. The content is initially cached at node N10. After a while, node N10 receives four Interests from consumers A, B, and C. To respond the consumers' Interests, node N10 becomes a provider and sends content C1 to consumers A, B, and C. In Figure 4.2, the consumer’s Interests are indicated by dotted line arrows. The solid-line arrows indicate the responses from the provider. Simultaneously, two Interests for content C2 are received from consumer D and consumer E at node N10. In response, N10 sends content C2 to consumer D and consumer E. According to the proposed caching mechanism, the total number of received Interests for content C1 is 6.

Subsequently, content C1 is selected as OPC because it has an extra Interest over the average of the total number of interests received for both content C1 and content C2. Therefore, the caching operations during the transmission of content C1 as OPC are done at mutually connected nodes N4 and N6 because both Node N4 and N6 are mutually connected with interested consumers as A, B and C. Therefore, the content C2 will only be cached at one mutual node as N11 that indicating lest distance from the interested consumers D and E as shown in Figure 4.2.

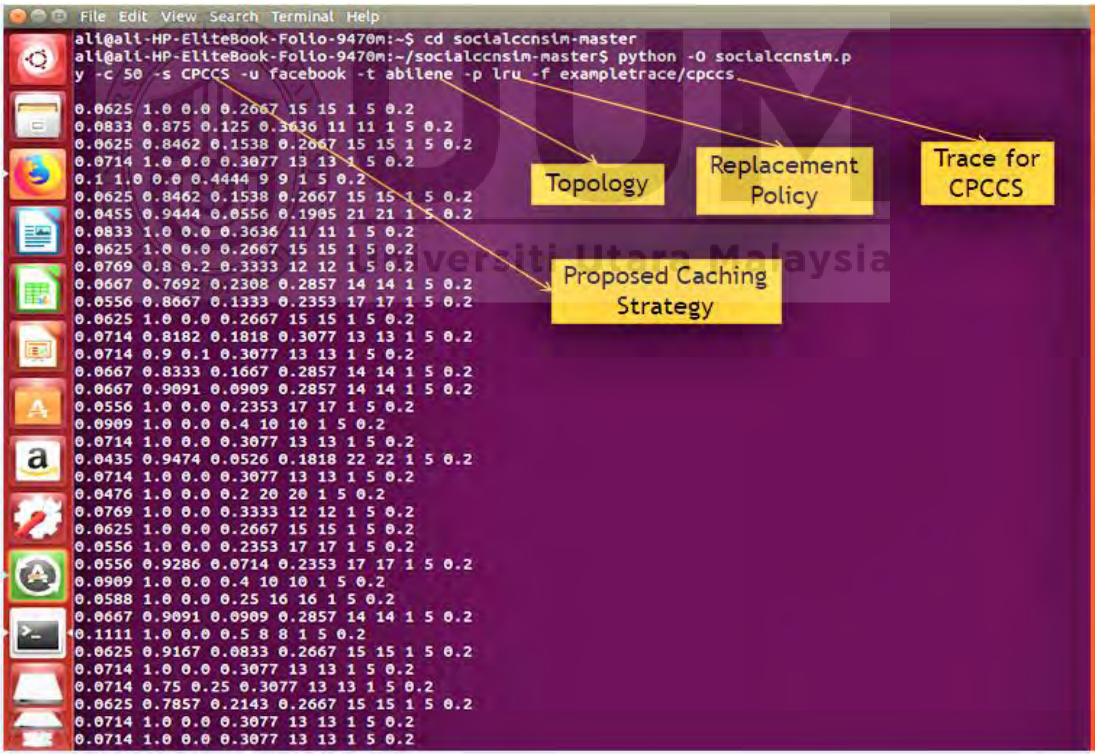

Figure 4.3. Verification of CPCCS through SocialCCNSim-Master

4.4 Verification of Compound Popular Content Caching Strategy (CPCCS)

It is essential to verify the proposed CPCCS caching strategy. The verification is done by considering all variables that are used to build the CPCCS model and logically check all the possible errors and remove the bugs form code. Therefore, it is verified from figure 4.1 that the proposed CPCCS is programmed correctly in SocialCCNSim-Master simulator. The code was written using python language and Ubuntu platform was considered to run the code and verified that the code is free from all the run time errors and bugs. Figure 4.1 illustrates the verified code of proposed CPCCS strategy is running using a new Trace file on Abilene topology.

4.5 Validation of Compound Popular Content Caching Strategy (CPCCS)

The CPCCS was developed by combining the two mechanisms such as Compound Popular Content Selection (CPCS) and Compound Popular Content Caching (CPCC). CPCCS model is verified using simulation platform (SocialCCNSim-Master) and the validation is done by the comparisons in which CPCCS caching performance is compared with other popularity-based caching strategies. Therefore, to evaluate the proposed content caching mechanism CPCCS, it is necessary to compare its attributes with analogous ones in earlier mechanisms [60, 102, 156]. Therefore, we selected the SocialCCNSim-Master simulator as the simulation environment for the performance comparison. SocialCCNSim-Master was specially designed to identify the performance of NDN caching. It takes data traffic from the Facebook social network topology, which is associated with 4,039 users who each have 44 relationships, and 88,234 friends.

SocialCCNSim-Master supports five ISP-level topologies (i.e., Abilene, Tree, GEANT, Tiger, and DTelecom), and each topology has its own structure. In this study, the Abilene topology was used for the simulation; it is broadly used because of its flexible structure and it forms a high-speed backbone by deploying cutting edge technologies. The objective of the Abilene topology is to transfer large amounts of data across the nodes. Its structure is perfect for testing and comparing our simulation parameters, recording the cache-hits, and observing the efficacy of CPCCS. Abilene topology has eleven (11) locations taken as nodes. Its hierarchical structure provides the flexible neighbor arrangement needed to test NDN content caching mechanisms.

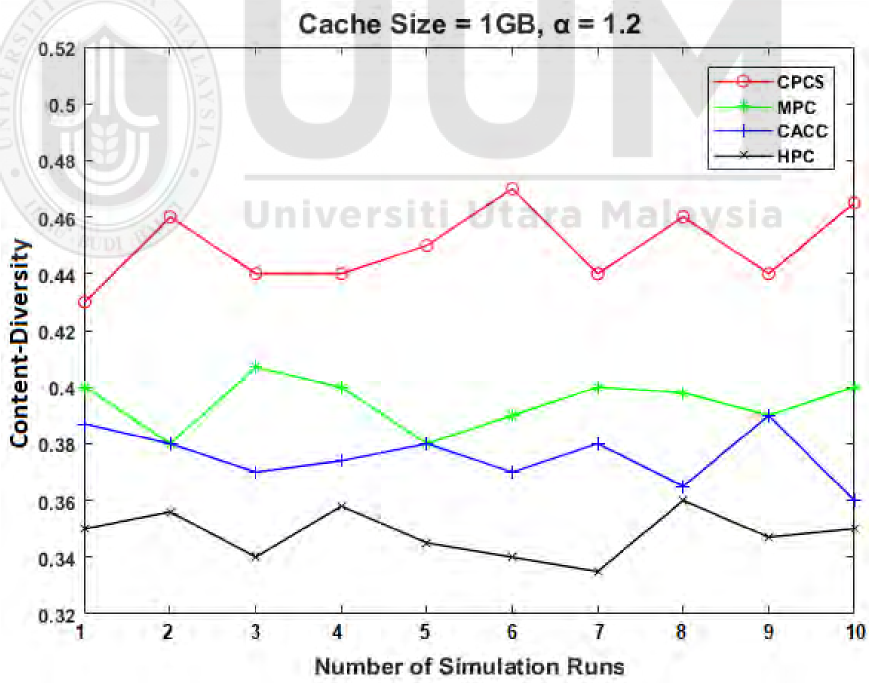

Figure 4.4. Content Diversity on Abilene Topology with 1GB cache size

To avoid unnecessary cache usage, the Least Replacement Policy (LRU) was used to replace the old content with the new. LRU is considered the most efficient content replacement policy because of its flexible structure and high performance. In this study, four different content categories were selected based on their popularity level, as defined through Zipf's distribution: User Generated Content (UGC), VoD, file sharing (file), and web content (web) [157]: we used these content categories because they showed high traffic production due to consumer interest in media-driven content. The Zipf content probability distribution function was used to select the content category. The cache size (i.e., the amount of space used to store content temporarily during its transmission) ranged from 1 GB to 10 GB, and the catalog size was10^8 elements of 10 MB each.

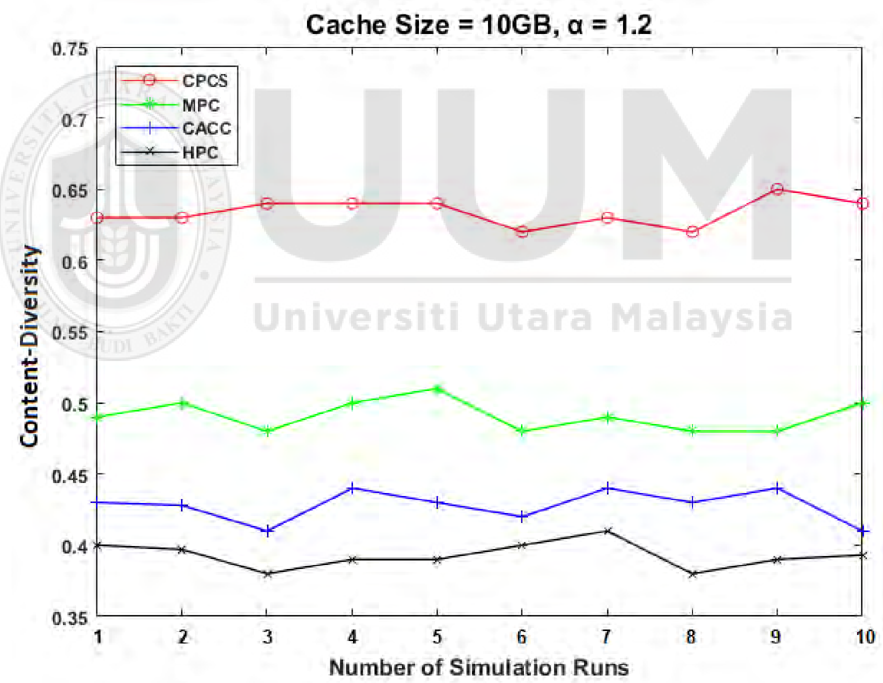

Figure 4.5. Content Diversity on Abilene Topology with 10GB cache size

The simulation scenario was divided into 10 equal steps, as shown with the x-axis in each simulation graph. The graphical representation of the consequences was achieved by taking the average result of all simulations.

Content diversity refers to the amount of heterogeneous content that is accommodated at certain locations. The purpose of the CPCS scenario is to minimize the high number of analogous content's replications to accommodate at a location. Moreover, the CPCS increases the amount of heterogeneous content in-network because its algorithm provides diverse content to be cached at the delivery path. HPC and CCAC performed poor in terms to achieve better diversity ration because both strategies generate multiple content replications along the data routing path. However, MPC performs better than both HPC and CCAC because it caches the contents only at neighbor routers.

From Figure 4.3 and Figure 4.4 it is concluded that CPCS performed much better because of its content selection mechanism. Therefore, it increases the number of diverse contents within the cache because it cached contents on the basis of their diverse popularities. In fact, CPCS is compared in a simulation environment to check the caching manageability whether its performance is better than MPC, CPCC, and HPC or not. Therefore, the simulation was run on Abilene topology with two different cache sizes to make comparison among the caching strategies to measure the number of diverse contents in the network. The contents are selected as VoD having Zipf value range about 1.2 regarding the recent studies. The consequences are shown in Figure 4.3 and Figure 4.4 with different cache size (1GB and 10GB respectively).

Content redundancy is referred to the multiple caching operations of similar contents. The higher the caching operation, the more possible more abundant redundant content would be cached along the data routing path.

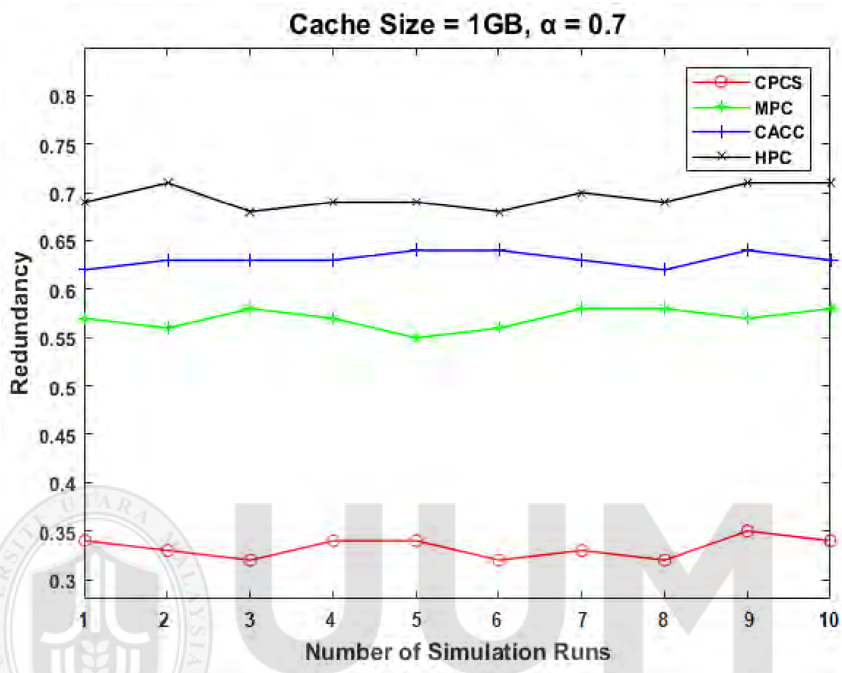

Figure 4.6. Content Redundancy on GEANT Topology with 1GB cache size

The result derived using GEANT topology to check the redundancy performance of different strategies as shown in Figures 4.5 and Figure 4.6. The cache size selected as 1GB and 10GB respectively and set the probability parameter as $\alpha = 0.7$ that is considered for file content category. From given (Figures 4.5 and 4.6), it is illustrated that the noticeable cause of similar frequent caching operations is made by HPC and CCAC due to its high-level usage of caches by homogeneous content replications, which reinforces content redundancy.

MPC shows the reduced redundancy because of its mechanism of caching content at limited locations that decreases the number of content replications as compare to HPC and CCAC. CICS performs better in terms of redundancy than HPC, CCAC, and MPC because of its nature to select diverse contents to be cached at homogeneous locations that increase the availability of diverse caching operations and reduces the replication of similar contents.

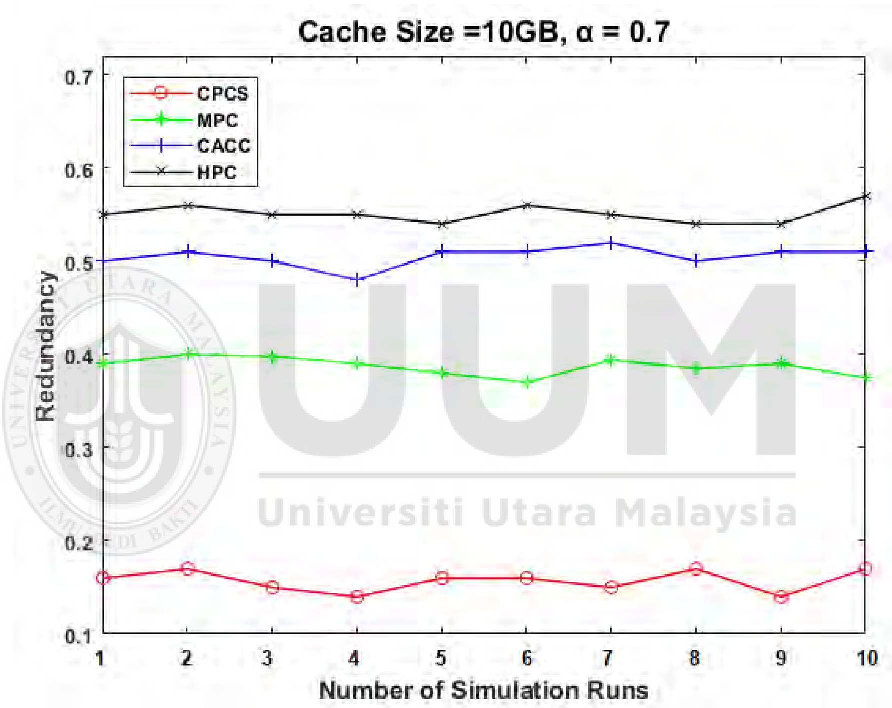

Figure 4.7. Content Redundancy on GEANT Topology with 10GB cache size

WAVE depicts lower cache hit ratio because it increases the number of similar contents along the data downloading path and it requires more time for caching entire chunks that belong to the same content and demonstrates a lesser cache hit ratio. Therefore, Interests for diverse content needed to be forwarded to the main provider, which makes for a long trip in content downloading. WAVE shows the larger path stretch with all cache sizes because it caches the popular content on the downstream node first, which increases the length between the consumer and provider. WAVE brings content close to the consumer, but it takes a long time because of its nature of caching decision.

For the validation of the CPCC mechanism, different caching strategies were selected to compare the performance analysis.

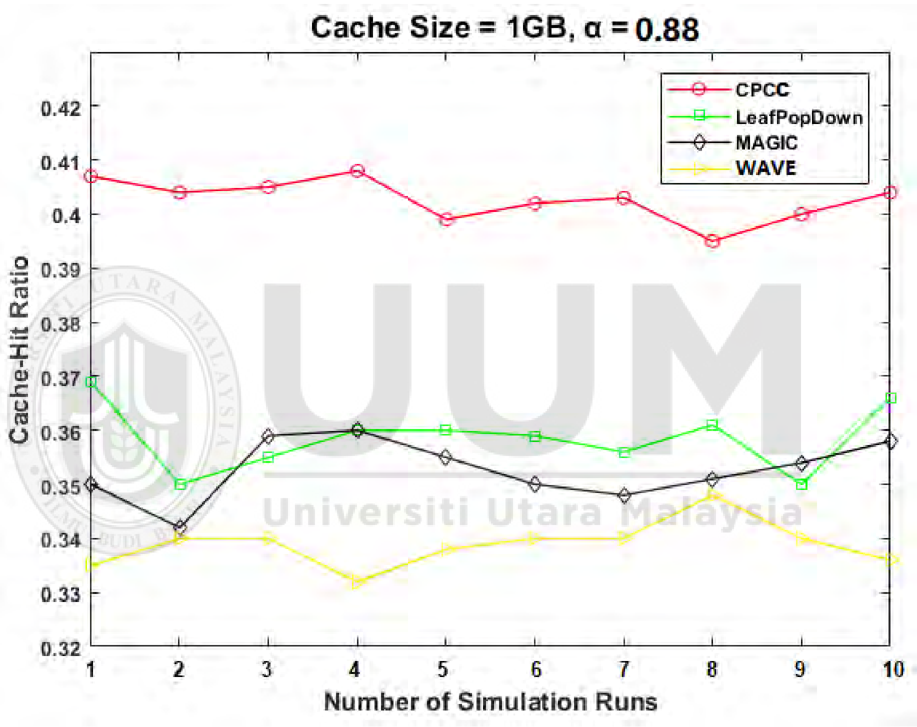

Figure 4.8. Cache Hit Ratio on DTelekom Topology with 1GB cache size

The cache hit ratio presents the quantity of the average of existing content hits (found) as the Interests are sent [135]. For a specific strategy, the cache hit ratio can be obtained by calculating the total number of interests and misses for nodes. For the benefit of comparison, the significance of the cache hit ratio is pictorially represented in Figure 4.7 and Figure 4.8. As cache size expands, the performance of the network improves noticeably, because the cache accommodates more contents within large cache size. The consequences that occur when the DTelekom topology is used while α = 0.8 was selected that is considered for User Generated Content (UGC). The CPCC performs better than comparing strategies because it prioritizes the desired content, which increases the chances to cache content near consumers, heightening the hit ratio.

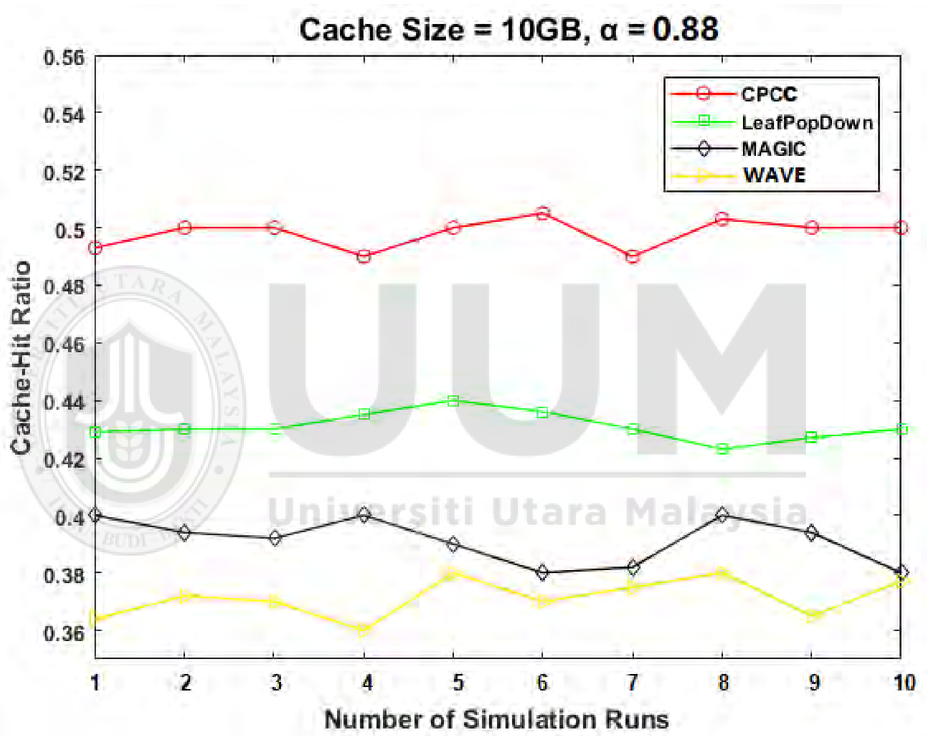

Figure 4. 9. Cache Hit Ratio on DTelekom Topology with 10GB cache size

Moreover, CPCC performed better than the benchmark caching mechanisms. CPCC showed good quality throughout the simulation results with UGC content categories. This is because CPCC caches OPC close to the consumers, which increases the availability of the most desired content for subsequent responses.

Another benefit of CPCC is that it initializes a time-span for each content, which indicates how long the content can be cached at a specific location (i.e., node). Therefore, CPCC decreases the unnecessary usage of cache storage and increases its ability to accommodate new content by removing unnecessary content. CPCC also improves the caching of heterogeneous content by selecting OPC and LPC.

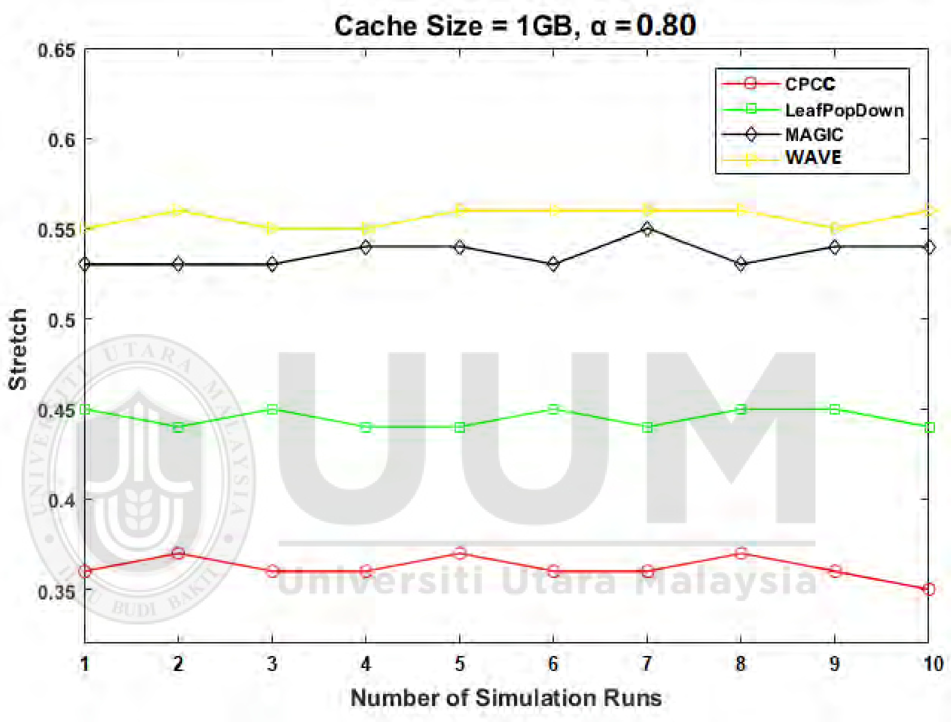

Figure 4.10. Cache Hit Ratio on Tree Topology with 10GB cache size

Stretch can be defined as the number of hops (routers) needed to be covered by a consumer's Interest between the source (where the hit occurs) and consumer. CPCC caches the popular content close to the consumers at the central position (mutual centrality node), from where all the desired consumers can get their required content. Therefore, it makes the distance between a consumer and a provider smaller and most of the consumer Interests travel through the central position and satisfies from mutual positions. Moreover, CPCCS selects diverse popular content to be cached close to the consumers that increase the overall stretch ratio. On the other hand, all the comparing strategies are showing the long stretch path as shown in Figure 4.9 and Figure 4.10.

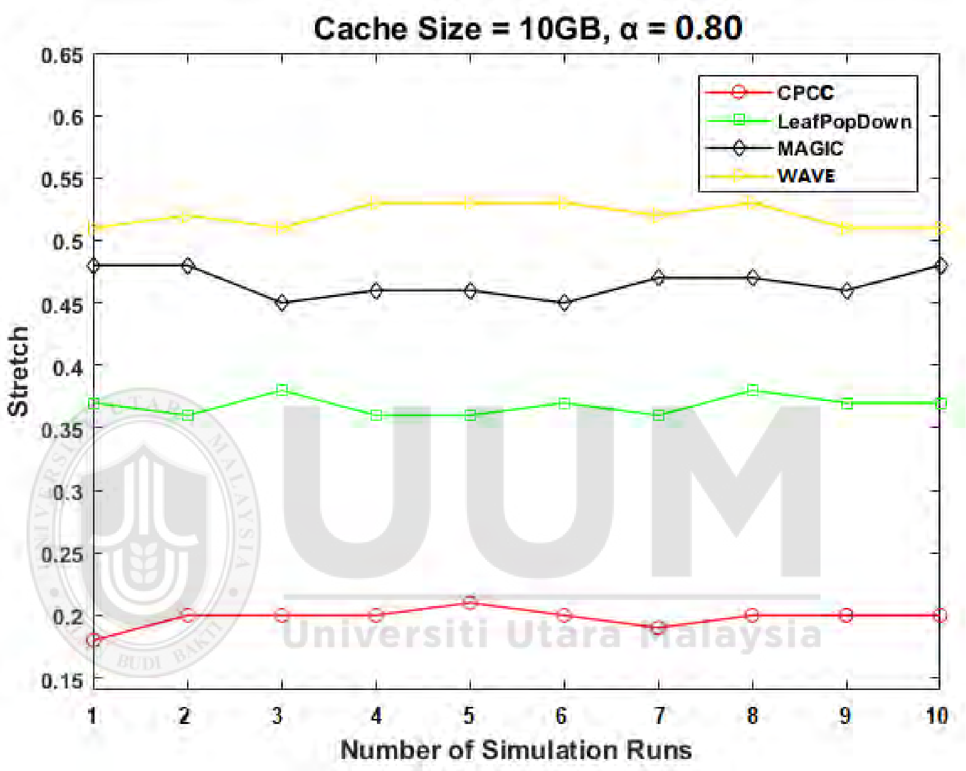

Figure 4.11. Cache Hit Ratio on Tree Topology with 10GB cache size

4.6 Summary

In this chapter, the content placement mechanism is explained and proposed the CPCCS model is described. The CPCCS is made up using two mechanisms in which CPCS separates the contents into OPC and LPC regarding the recent utility of contents and CPCC is proposed for the caching of requested content close to the appropriate consumers. The CPCS is developed to increase the selection of diverse content for caching along the data routing path and it reduces the number of redundant replications of similar contents. On the other hand, the CPCC mechanism reduces the stretch path length by caching the content near the consumers and consequently, the cache hit ratio is increased. The CPCCS is verified by adding its codes to SocialCCNSim-Master simulator and validate it in simulation platform using different topologies with diverse content categories through comparisons between the performances of CPCCS and MPC, CCAC, HPC, MAGIC, WAVE, LeafPopDown. Therefore, the consequences show the CPCCS performed better in term of content diversity, content redundancy, cache hit ratio, and stretch ratio.

CHAPTER FIVE
SIMULATION RESULTS AND EVALUATION

5.1 Introduction

NDN caching is a revolution in network architecture. It can overcome the caching issues that arise in the current Internet architecture. Moreover, it can reduce communication overhead, resource consumption, and bandwidth, through caching popular content at multiple locations. However, it is difficult to decide which content needs to be cached at a location to produce optimal solutions. Cache-management strategies [60, 78, 156] have been developed to achieve efficient results. Still, it is not clear which caching mechanism is the most ideal for each situation. Even though many methods have been proposed as described above, the basic issue still faced by all mentioned caching strategies is as follows. If a content is located far from the desired consumer, and numerous Interests are received for that content, but the popularity of that content is lower compared to the popular content, the content will not be suggested as a popular content and all the times, the Interests for the content need to traverse through the several hops to find the required content. Therefore, all these strategies focus on the most popular or most recently interested content that cannot fulfill the caching requirements for less popular contents.

Consequently, the overall caching performance is disturbed in terms of caching hit ratio, stretch, redundancy, and diversity. Therefore, the proposed caching mechanism in this paper is more flexible to handle the issues of caching. The proposed caching strategy not only prefers the most frequently interested content but also handles the less popular contents to increase the overall caching performance.

The proposed CPCCS cache deployment strategy was introduced previously in chapter four. Basically, CPCCS made up by the combination of two mechanisms such as CPCS and CPCC. Moreover, the verification was presented using a simulation platform with varying parameters and topologies. However, this chapter presents the evaluation of CPCCS through comparisons between its performance and other popularity-based caching strategies.

Section 5.2 represents the performance evaluation in which parameters for simulation are defined such as topology, content popularity model, and cache size. In section 5.3, the simulation and numeric results are explained using graphical and tabular form. In section 5.4, chapter summary is presented.

5.2 Performance Evaluation

To compare and evaluate the performance of the mentioned caching strategies in chapter two related study, a caching-based simulator called SocialCCNSim-Master was selected. For the evaluation of the proposed CPCCS strategy, we conducted a simulation environment using SocialCCNSim-Master simulator [102, 156]. The caching performance of the proposed strategy in terms of content diversity, cache-hit ratio, content redundancy, and stretch is evaluated and compared with the caching performance of HPC, MPC, LeafPopDown, CACC, WAVE, and MAGIC caching strategies. For content replacement operations, most of the NDN caching strategies adopt the Least Recently Used (LRU) replacement policy [78]. LRU policy is the optimal content replacement policy due to its better performance in terms of overhead and complexity. Due to NDN's cache-driven architecture, various algorithms and strategies have been designed to choose from based on the set of rules regarding which content to cache from the varieties of popular content, content-defined names, and topological content information, among others. For the efficient evaluation, different categories of contents were selected for caching operation based on their popularity as defined through Zipf's distribution [79]: user-generated content (UGC), Video on Demand (VoD) content, web content, and file content [80]. The present study used Zipf 0.7, 0.80, 0.88, and 1.2 for file content, UGC content, web content, and VoD content respectively. These categories were selected because they show high traffic production due to consumer interest in media-driven content [158].

In the study, CPCCS was examined the performance using Abilene, GEANT, Tiger, and DTelekom topologies [159]. The traffic was generated by SONETOR and using social network graph as Facebook [160]. The catalog size was selected as 10^8, cache size (1GB to 10GB), and the simulation was run for 24 hours to create results for each result.

5.2.1 Network Topology

The NDN caching strategies have been tested and evaluated using different topologies. For instance, Chai et al used tree topology with 4 to 6 levels k-ary in which all the nodes further have 2 or 5 children, Li et al. considered 4-level tree topology, Psaras et al selected 6-level binary tree topology, Rossi et al used five different topologies such as; GEANT, Tiger, Abilene, DTelekom, and Level3. For the NDN caching evaluation, Bernardini et al, Din et al, and Naeem et all were used ISP-level topologies such as GEANT, Tiger, DTelecom, and Abilene. For the evaluation of NDN caching strategies, it is significant to choose an appropriate network topology because topology is influenced directly on simulation results. Therefore, fair evaluation needs some suitable network topology to generate true results. Thus, to evaluate the proposed caching strategy four ISP-level topologies such as Abilene, GEANT, DTelekom, and Tiger were selected. Abilene formed a high-speed backbone by deploying cutting edge technologies not yet generally available on the scale of a national network backbone. Abilene was a private network used for education and research but was not entirely isolated, since its members usually provide alternative access to many of their resources through the public Internet. Sometimes it referred to Intetnet2 network topology. Abilene topology was designed with the sole object of transferring large size data across nodes. Its building was ideal to test and compare our simulation parameters to record the cache-hits and observe the efficacy of our cache managements. Abilene heterogeneous nature depicts eleven (11) station sites as nodes. Its hierarchical structure provides the flexible neighbor composition needed to test NDN cache deployment strategies. This forms the basis of our topology selection to test the performance of the strategy for the heterogeneous Internet.

On the other hand, the GEANT topology was developed especially for research, education, and innovation communities across the globe. It is pan European network that is used to interconnect Europe's National and Education Networks (NRENs). GEANT has the ability to connect more than 50 million consumers at 10,000 organizations. Moreover, GEANT provides high bandwidth and secure high capacity (50,000km) network with an increasing range of services that allow the researchers to cooperate with one another wherever they are located. GEANT and Tiger topologies have 22 nodes and DTelekom consist of 68 nodes contented to each other as shown in Figure 5.1.

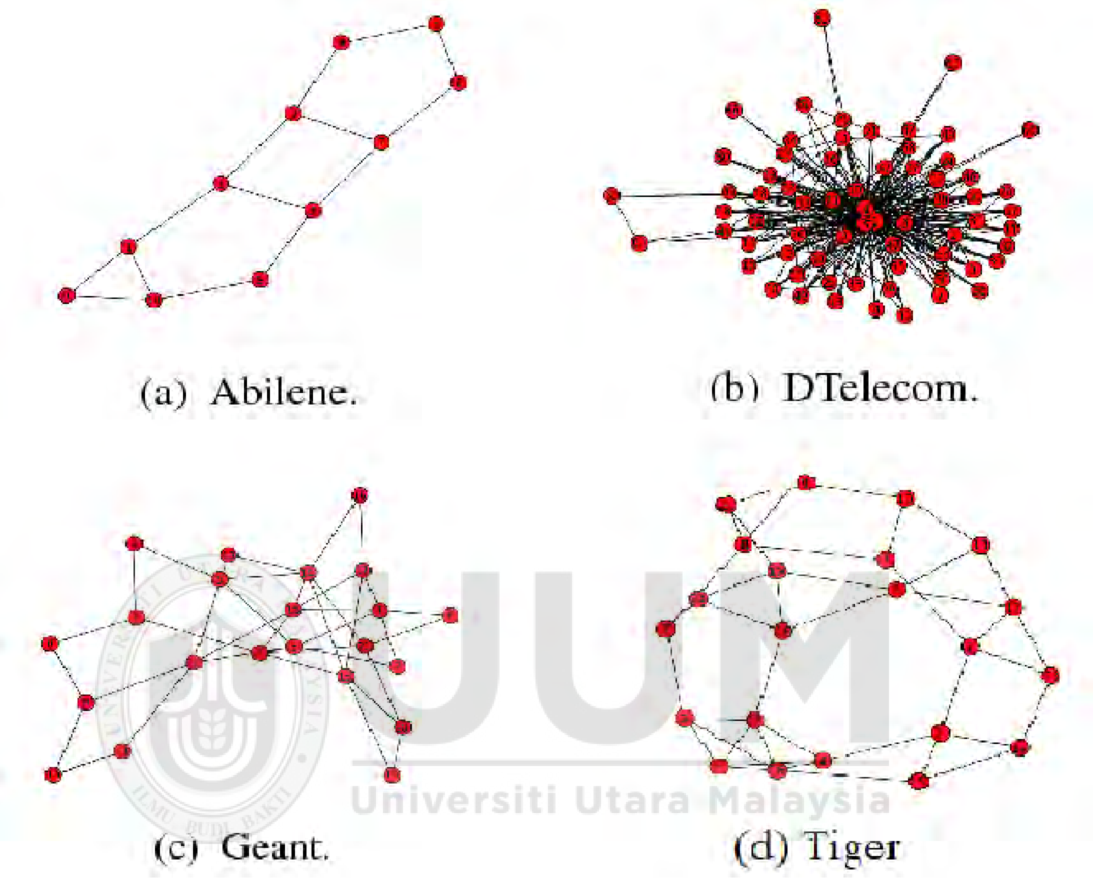

Figure 5.1. ISP-Level Topologies

5.2.2 Content Popularity Model

The content popularity model is used to consider the utility of contents in terms to estimate the popularity for individual content. In fact, it is a function that calculates the number of Interests received for a single content. According to former research, there have been millions of websites deployed across the globe but a little number of websites is attracted by the majority of the consumers. It is important to categories the content according to their requesting frequency. Hence, the popularity can be achieved by calculating the on-demand frequency for a specific category of content. According to recent studies, the power [161, 162] law for content popularity distribution is the most significant to define content popularity. According to a power law, Zipf distribution is an ideal solution that is commonly used in simulation models for categorization of content. Therefore, it is most appropriate for this study to select the Zipf model for simulation to set the popularity parameter.

$$pmf(x, N, q, \alpha) = \frac{(q+x)^{-\alpha}}{\sum_{k=1}^{N}(q+x)^{-\alpha}} \tag{5.1}$$

In equation 5.1, the alpha α parameter represents the popularity distribution slop, N represents the number of content items, q shows the placement, and pmf describes the probability mass function. Numerous researches have been used Zipf approach for the evaluation of their proposed model in which Saino *et al,.* Zhang *et al*, and Li et al used Zipf popularity model with different values of the alpha α parameter as 0.6 to 1.1, 0.3, and 0.9 respectively. In ICN, the recent studies Bernardini *et al* and Fricker *et al.* selected Zipf model with α parameter as 0.6 to 2.5. In a similar study of CCN, Carofiglio et al. are used Zipf distribution with α parameter as 1.6 to 2.0. Similarly, in [163] and [164] the α parameter was taken as 0.8 and 0.5 to 2.5 respectively. Recently, in [164], [165] and [165] the Zipf distribution model is used with α parameter as $\alpha = 1, \alpha = 1\ .2,$ and $\alpha = 0.5$ to $\alpha = 1.5$ respectively. In [166], the Carlinet *et al.* was suggested Zipf popularity distribution for UGC service DailyMotion with alpha parameter as $\alpha = 0.88$. Moreover, Carlinet *et al.* also estimate the Zipf popularity distribution for VoD with alpha parameter as $\alpha = 1.2$. On the other hand, the study [167] by Yu *et al.* Zipf popularity distribution with alpha from .65 to 1 was taken for VoD service in china.

Therefore, alpha value has a great influence on the overall performance and provides diversified content categories for suitable evaluation. Hence, this research is selected the Zipf distribution with distinguished content popularity sets named as File content, Web content, User Generated Content (UGC), and Video on Demand (VoD) with alpha α value as 0.7, 0.80, 0.88, and 1.2 respectively. According to the recent report of Cisco Visual Network Index (VNI), there is a high demand for information every minute. Now, the consumers are largely interested in multimedia-driven content rather than the ordinary file transfer. Diverse Internet consumers have a tendency for multimedia data contents and high demand for shared videos on YouTube that have recorded as 139,000 hours of videos watched which is approximately 300 hours of videos are uploaded in each minute. The total videos of YouTube and NETFLIX are approximately equal to the half traffic on the Internet.

5.2.3 Cache Size

Cache size refers to temporary storage space available to provide room for transmitted content within each NDN node in a network. The cache is used as temporary storage because of its limited capacity available in physical devices (i.e., network router). Usually, the cache is flexible and it is available with different size to achieve an efficient performance. Several NDN caching-based studies recently been developed in which Zhang *et al*. was selected cache size as 1GB while Cesar [102], and Ibrahim *et al.*, Din *et al.* were selected cache size as 100 to 1000 elements in which 100 represent the cache size is 1GB and 1000 shows the cache size as 10GB. Moreover, Dabirmoghaddam *et al.* [113] and Saino [140] have chosen cache sizes as multiples of 10 (e.g., 10^2, 10^3). In addition, Fricker *et al.* [168] and Bilal *et al*. [169] expressed the cache size in the form of elements such as 100, 200, 500, 1000, 10000. In order to achieve efficient performance it significant to adopt cache size as 1GB to 10 GB in which each size represents the number of elements as 100 to 1000 respectively. Moreover, each element occupies 10MB [168, 169] cache space within the network node during its transmission between sender to receiver.

5.3 Simulation Setup

To compare and evaluate the performance of the proposed CPCCS and MAGIC, WAVE, HPC, LeafPopDown, MPC, CCAC caching strategies, a caching-based simulator called SocialCCNSim-Master is used. The performances were compared in terms of the cache hit ratio, content diversity, stretch, and content redundancy of popularity-based caching strategies. For content replacement operations, most of the NDN caching strategies adopt the Least Recently Used (LRU) replacement policy [170]. LRU policy is the optimal content replacement policy due to its better performance in terms of overhead and complexity. Due to NDN's cache-driven architecture, various algorithms and strategies have been designed to choose from based on the set of rules regarding which content to cache from the varieties of popular content, content-defined names, and topological content information, among others. For the purposes of this study, the research will caching sets of content of four categories based on their popularity defined through Zipf's distribution [171].

web content, file sharing, user-generated content (UGC), and VoD [172]. Table 5.1 shows the basic parameters selected for the current simulation.

Metrics have been used to test the effectiveness of cache management strategies. For the present study four well-known metrics content diversity, cache hit ratio, redundancy, and stretch were selected to compare the performance of the proposed caching strategy.

Table 5.1

Simulation Description Values.

Parameter	**Value/Description**
Simulator	SocialCCNSim-Master
Traffic Source	SONETOR (Facebook)
Content Categories	File, Web, UGC, and VoD
Simulation time	24 hours
Chunk Size	10 MB each
Cache Size	1 GB to 10 GB (100-1000 elements)
Catalog Size	10^8
Topology	Abilene, GEANT, DTelekon, and Tiger
Replacement Policy	Least Recently Used

In each simulation graph, the x-axis is split into 10 equal sections that represent the incremental cache size. Each section represents with an increment of 1GB (100 elements) as starting from 100 to 1,000. While the y-axis shows the percentage value of achieved results.

5.3.1 Content Diversity

It is the ratio of heterogeneous contents accumulated in-network caches [60, 173-175]. Diversity can be defined as follows:

$$D = \frac{\bigcup_{v=1}^{n} X_v}{\sum_{v=1}^{V} C_v} \tag{5.2}$$

where $\bigcup_{v=1}^{V} X_v$ represents the amount of unique content within the n number of nodes, and $\sum_{v=1}^{V} C_v$ shows the cache storage of all nodes in a network. V represents the total number of nodes.

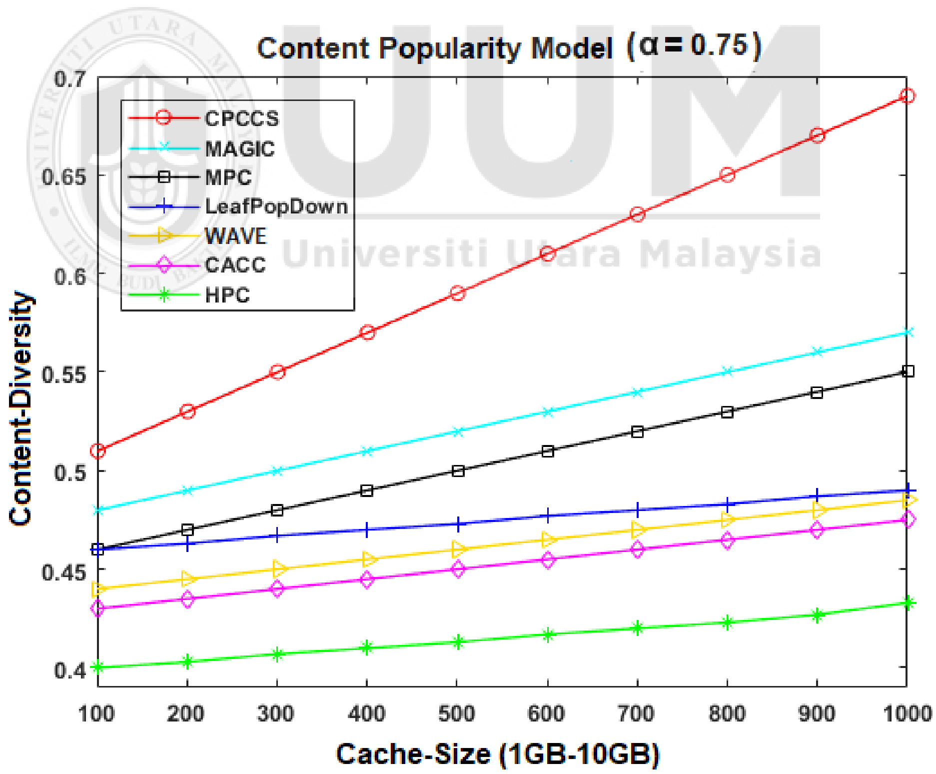

Figure 5.2. Content Diversity on Abilene topology (File content)

The intention of the proposed caching strategy is to reduce the high amount of similar-content replications and to make room for new incoming contents. The replication of similar content occupies a large amount of cache storage, which increases the amount of network traffic and congestion and minimizes the amount of diverse content within the network [176]. HPC shows the lowest diversity, which means that the replication of homogeneous contents in HPC is higher than other strategies. From the simulation results shown in Figure 5.2, Figure 5.3, Figure 5.4, and Figure 5.5, we can conclude that the HPC caching strategy shows less diversity due to its algorithm leaving all the transmitting contents everywhere. Moreover, when we enlarge the cache size from 1GB to 10GB, HPC achieves slightly better diversity ratio with all content categories.

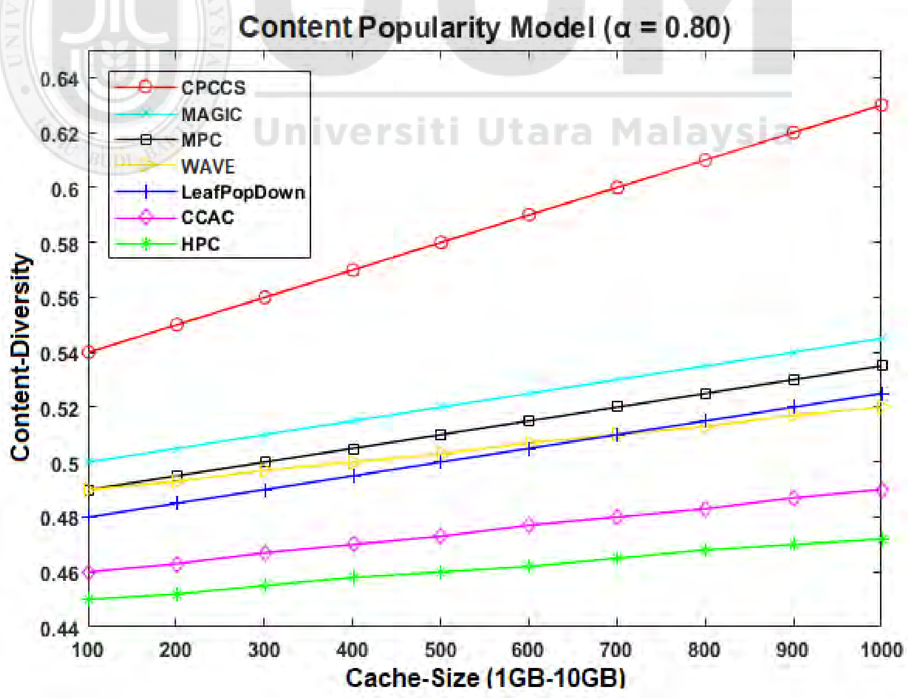

Figure 5.3. Content Diversity on Abilene topology (Web content)

The reason is that it does not reduce the replications done by similar contents within a large cache size. In the same way, it shows less quantity of diversity than all comparing strategies as shown by numeric values in Table 5.2. CCAC shows higher diversity ratio than HPC because it reduces the homogeneous content replications than HPC. However, it shows little diversity due to the higher amount of similar replications of content than MPC. As compared to other strategies, MPC performs in a different way to some extent, because of its popularity method of caching content.

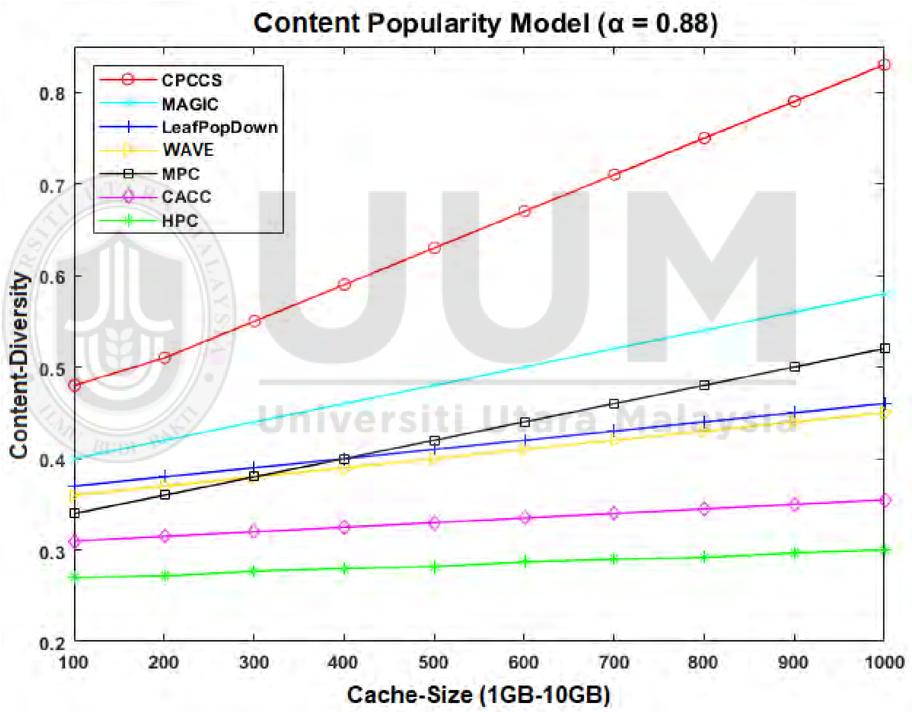

Figure 5.4. Content Diversity on Abilene topology (UGC content)

MPC performs better with large cache size than other strategies because it allows to caches content only at neighbor nodes and remains the other nodes empty along the data routing path. As a result, with large cache size, supplementary content can be accommodated with diverse nature.

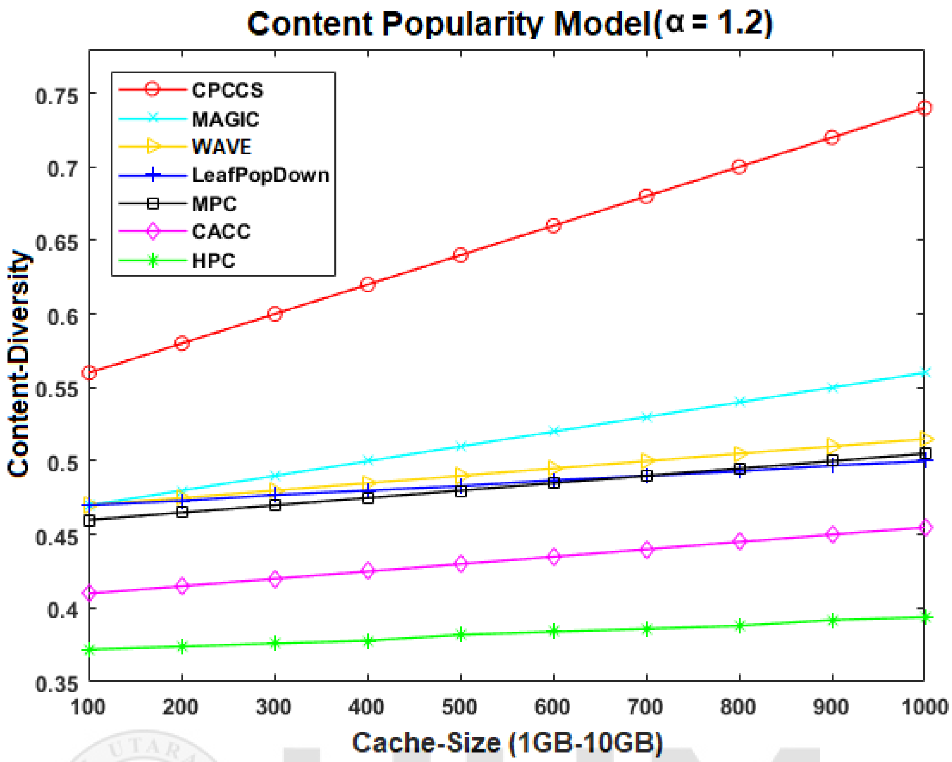

Figure 5.5. Content Diversity on Abilene topology (VoD content)

Table 5.2

Content Diversity numeric results on Abilene

Abilene Topology	**File content ($\alpha = 0.75$)**		**Web Content ($\alpha = 0.80$)**		**UGC Content ($\alpha = 0.88$)**		**VoD Content ($\alpha = 1.2$)**	
Strategies	1GB	10GB	1GB	10GB	1GB	10GB	1GB	10GB
HPC	0.40	0.44	0.45	0.47	0.28	0.30	0.37	0.39
CACC	0.43	0.47	0.46	0.49	0.31	0.35	0.41	0.46
WAVE	0.44	0.48	0.49	0.51	0.37	0.46	0.47	0.53
MPC	0.46	0.55	0.49	0.54	0.35	0.52	0.46	0.51
LeafPopDown	0.46	0.49	0.48	0.52	0.38	0.47	0.54	0.52
MAGIC	0.48	0.53	0.50	0.55	0.40	0.59	0.47	0.57
CPCCS	0.51	0.68	0.54	0.62	0.49	0.81	0.56	0.73

LeafPopDown and WAVE achieve moderate results due to their nature of algorithms to cache content at partial nodes. The reason is that, both strategies allow the limited number of replications. MAGIC shows better performances in terms of diversity because it allows limited content's replications along data delivery path and it increases the number of diverse content to be cached at a location. However, CPCCS boosts the diversity ratio because it does not allow similar content to be cached at multiple locations. When we enlarged the cache from 1GB to 10GB, CPCCS still performs better than the other strategies because it does not allow content to be replicated at numerous locations within large cache size. Indeed, CPCCS allows limited copies of the content to be cached along with the data delivery path. If the path is associated with a short stretch, the CPCCS will caches content only at unique nodes.

Indeed, it is concluded from the simulation results in Table 5.2 CPCCS is achieved better performance with all content categories (File, Web, UGC, and VoD) in terms to improve diversity as compared to other strategies.

5.3.2 Cache Hit Ratio

The cache hit ratio is the key metric in evaluating the performance of the NDN cache. It refers to the response by the in-network cache storage in which the content is locally cached for a specific time period [177]. It can be defined as follows:

$$Cache\ Hit\ Ratio = \frac{\sum_{n=1}^{N} hit_i}{\sum_{n=1}^{N}(hit_i + miss_i)} \qquad (5.3)$$

The cache hit occurs when a consumer's required content is found at the network node's cache. Therefore, the cache reacts as a provider by sending the corresponding content to the appropriate consumer [170].

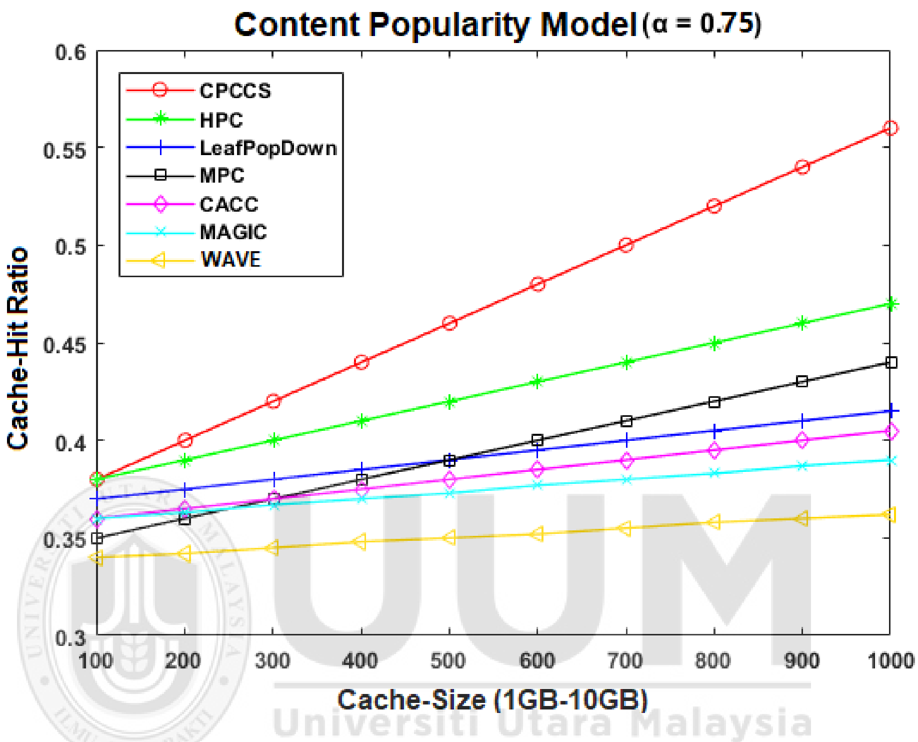

Figure 5.6. Cache Hit Ratio on GEANT topology (File content)

Figure 5.6, Figure 5.7, Figure 5.8, and Figure 5.9 shows the results achieved for the cache hit ratio, demonstrating that CPCCS performed better than the benchmark caching mechanisms. CPCCS showed good quality throughout the simulation results in all content categories (i.e., File, Web, VoD, and UGC). This is because CPCCS caches OPC close to the consumers, which increases the availability of the most desired content for subsequent responses. Another benefit of CPCCS is that it initializes a time-span for each content, which indicates how long the content can be cached at a specific location (i.e., node). Therefore, CPCCS decreases the unnecessary usage of cache storage and increases its ability to accommodate new content by removing unnecessary content. CPCCS also improves the caching of heterogeneous content by selecting OPC and LPC.

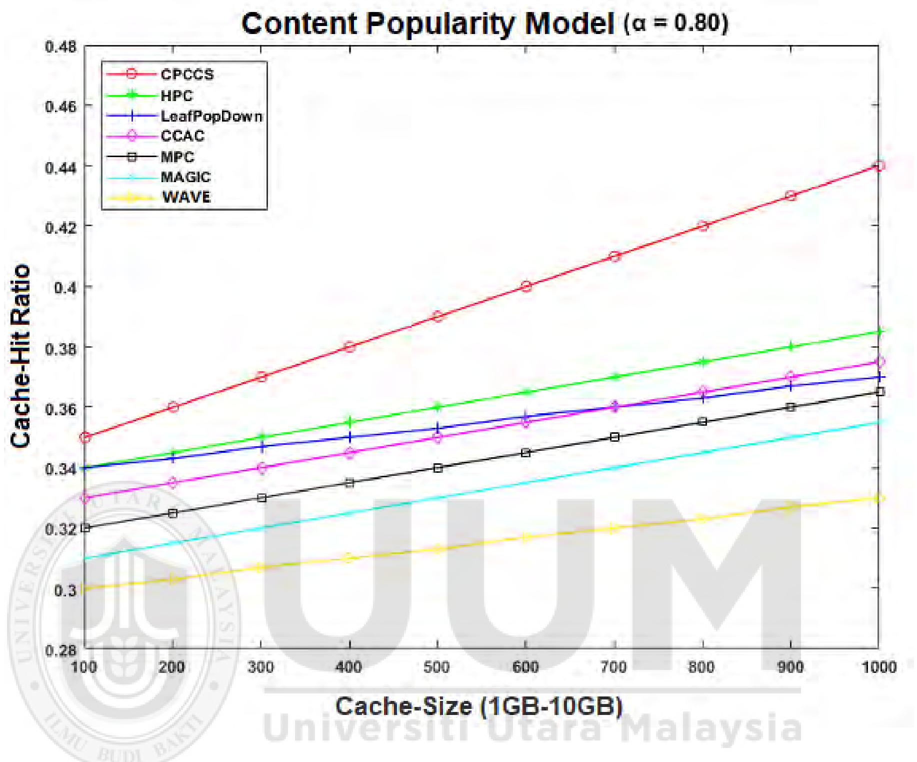

Figure 5.7. Cache Hit Ratio on GEANT topology (Web content)

WAVE, MAGIC, LeafPopDown, and HPC show similar performance in terms of the cache hit ratio with small and large cache size. However, HPC shows better cache-hit ratio due to its selection of caching content at all on-path nodes for a specific time span. Meanwhile, MPC performed better than WAVE and CACC, but its results were not as favorable as those of CPCCS because MPC took longer to select popular content. MPC uses a popularity table for each content, which increases the searching overhead when calculating which content is the most popular.

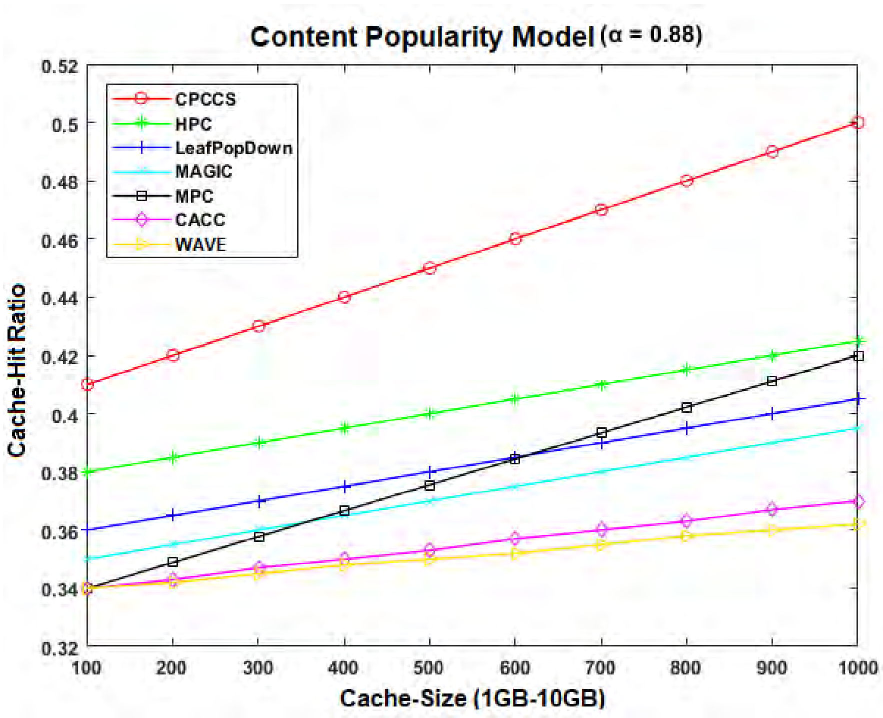

Figure 5.8. Cache Hit Ratio on GEANT topology (UGC content)

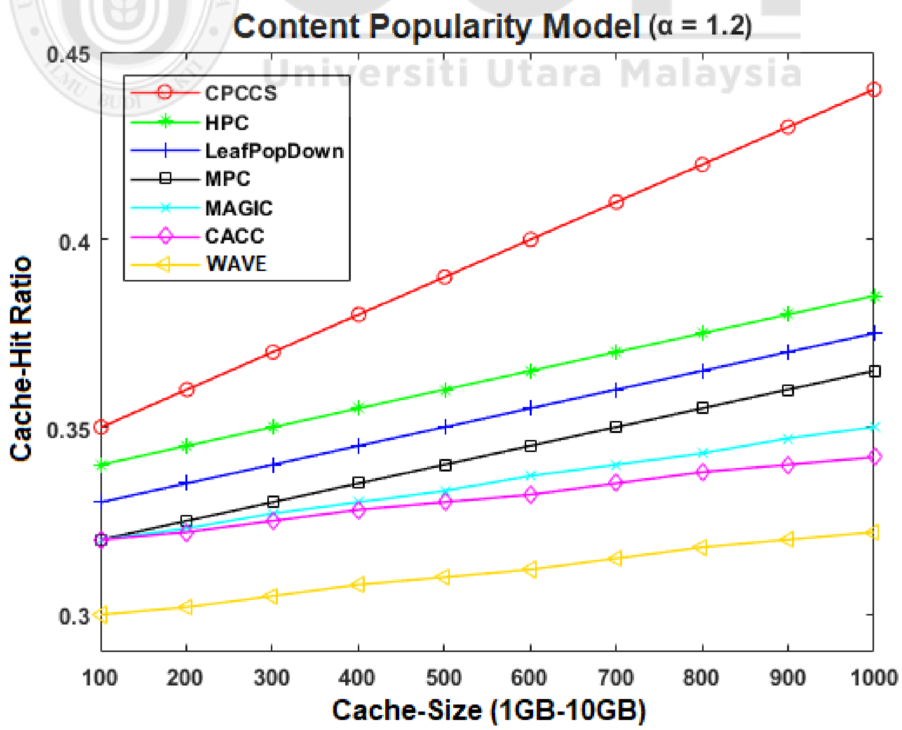

Figure 5.9. Cache Hit Ratio on GEANT topology (VoD content)

Moreover, MPC cached the most popular content at all neighbor nodes, which increased the unnecessary usage of cache storage and lessened its chances of accommodating new incoming content; thus, the overall hit ratio was decreased. CACC performed slightly worse than MPC because it cached all of the content regardless of the consumer interest level at the intermediate node, increasing the chance of accommodating the least popular content. WAVE depicts lower cache hit ratio because it increases the number of similar contents along the data downloading path and it consumes extra time to bring the required content near the consumers.

Table 5.3

Cache Hit Ratio numeric results on GEANT

GEANT Topology	**File content ($\alpha = 0.75$)**		**Web Content ($\alpha = 0.80$)**		**UGC Content ($\alpha = 0.88$)**		**VoD Content ($\alpha = 1.2$)**	
Strategies	1GB	10GB	1GB	10GB	1GB	10GB	1GB	10GB
CPCCS	0.38	0.56	0.35	0.44	0.41	0.50	0.35	0.44
HCP	0.38	0.47	0.34	0.38	0.38	0.43	0.34	0.38
LeafPopDown	0.37	0.42	0.34	0.37	0.36	0.41	0.33	0.37
CCAC	0.36	0.41	0.33	0.38	0.34	0.37	0.32	0.34
MAGIC	0.36	0.39	0.31	0.36	0.36	0.39	0.32	0.35
MPC	0.35	0.44	0.32	0.37	0.34	0.42	0.32	0.37
WAVE	0.34	0.37	0.30	0.33	0.34	0.36	0.30	0.33

Therefore, Interests for diverse content need to be forwarded to the main provider, which makes for a long trip in content downloading. When the cache size is increased, CPCCS again shows better results as the cache hit ratio increases because of its nature in caching heterogeneous content close to the consumers. From the numerical results shown in Table 1, we may conclude that CPCCS performs better in terms of the cache hit ratio than the comparing strategies.

5.3.3 Content Redundancy

Content-redundancy shows the amount of similar content caching operations at multiple locations in the same network [178-180]. It can be calculated using the following equation:

$$Redundancy = \sum_{i=1}^{n} Rc_i \tag{5.4}$$

where Rc_i shows the redundancy given by the i[th] item of the cached contents. The result in Figure 5.10, Figure 5.11, Figure 5.12 and Figure 5.13 illustrates content-redundancies on DTelekon topology.

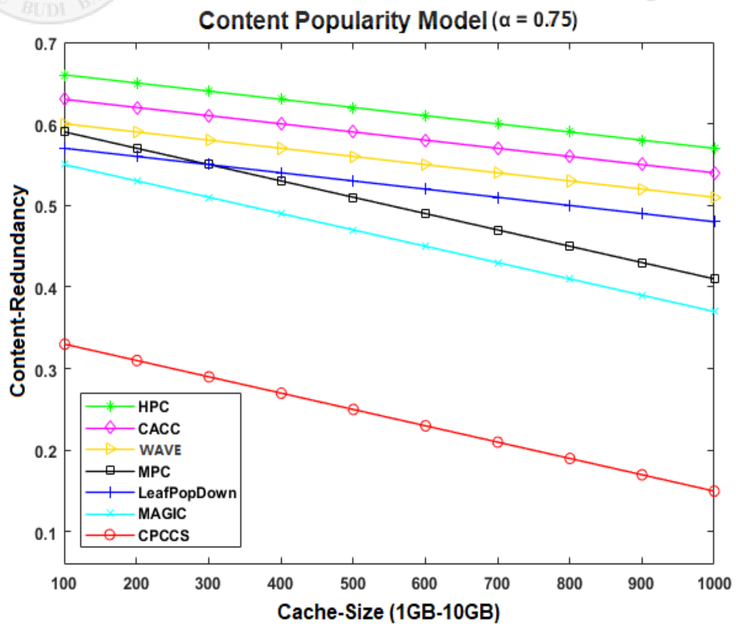

Figure 5.10. Content Redundancy on DTelekom topology (File content)

HPC and CACC show higher content-redundancies because both strategies use to cache all the contents at all the nodes at the same path that increases the high amount of similar content duplications. MPC, WAVE, and LeafPopDown show fewer repetitions of similar contents than HPC and CACC because MPC caches the content at neighbor nodes and WAVE perform caching operation of content after having a large number of Interests received. Moreover, WAVE consumes more time to caches content at all the nodes because it delivers the content in distributed chunks forms that gradually pushes towards the consumers.

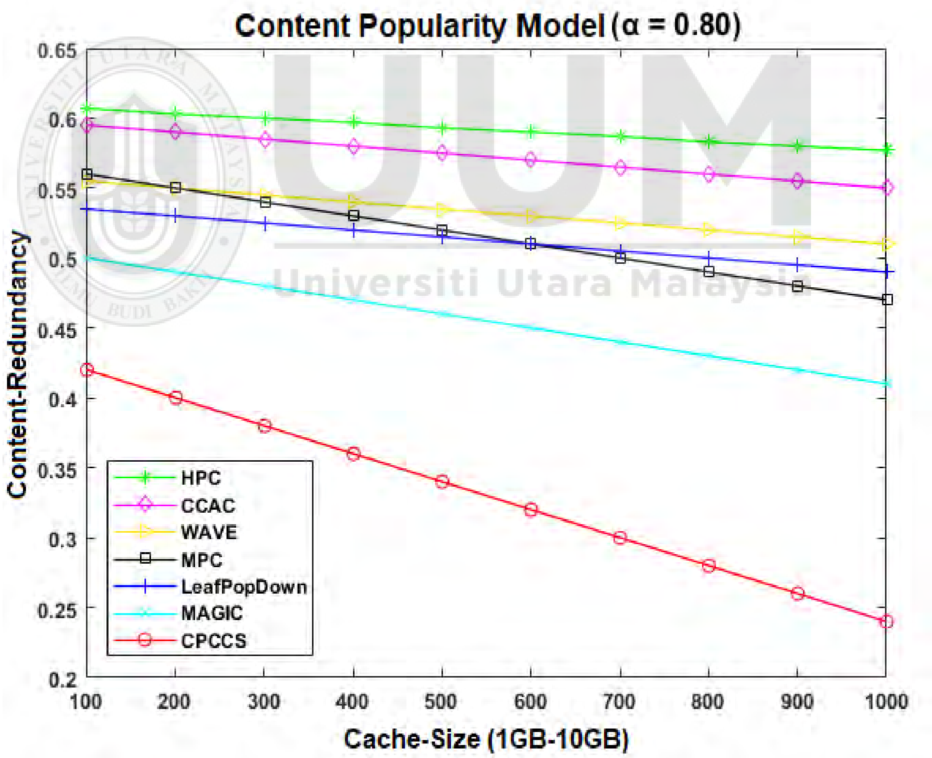

Figure 5.11. Content Redundancy on DTelekom topology (Web content)

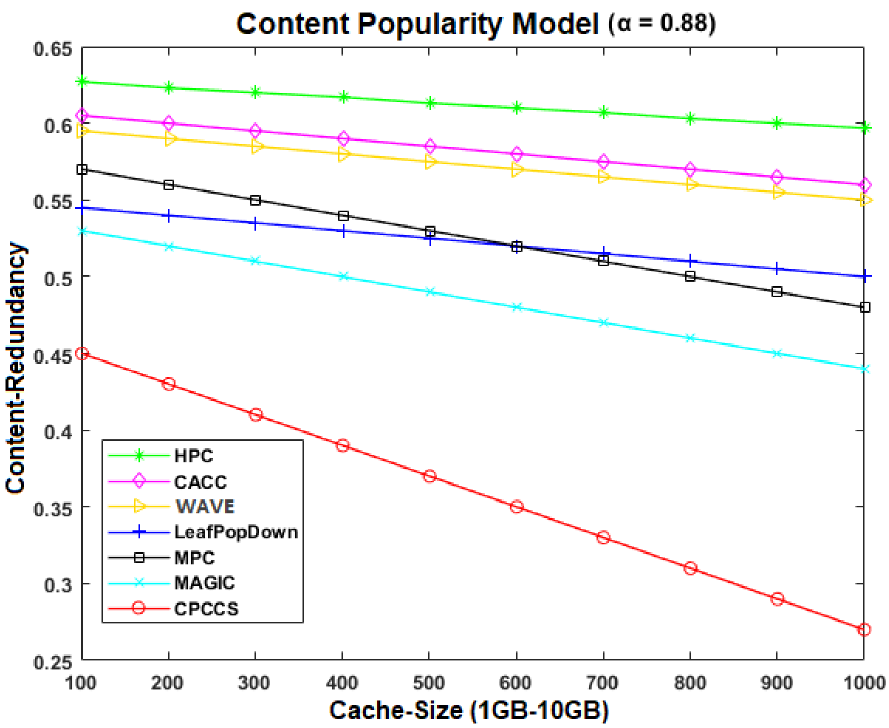

Figure 5.12. Content Redundancy on DTelekom topology (UGC content)

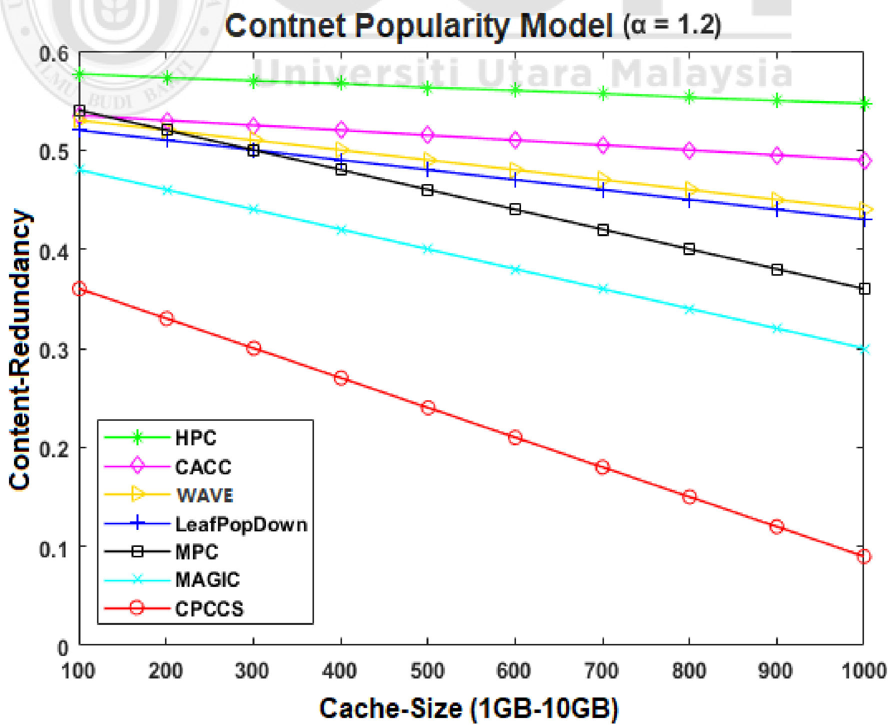

Figure 5.13. Content Redundancy on DTelekom topology (UGC content)

In addition, LeafPopDown caches content at the edge and downstream nodes. Therefore, the subsequent are satisfies from the edge node to some extent that decreases the redundant caching operations by similar contents.

Table 5.4

Content Redundancy numeric results on DTelekom

DTelekom Topology	**File content ($\alpha = 0.7$)**		**web Content ($\alpha = 0.80$)**		**web Content ($\alpha = 0.88$)**		**web Content ($\alpha = 1.2$)**	
Strategies	1GB	10GB	1GB	10GB	1GB	10GB	1GB	10GB
HPC	0.68	0.59	0.67	0.61	0.67	0.61	0.67	0.61
CACC	0.64	0.55	0.60	0.56	0.60	0.56	0.60	0.56
WAVE	0.61	0.51	0.57	0.53	0.57	0.53	0.57	0.53
MPC	0.59	0.41	0.59	0.51	0.59	0.51	0.59	0.51
LeafPopDown	0.58	0.48	0.54	0.49	0.54	0.49	0.54	0.49
MAGIC	0.57	0.38	0.55	0.47	0.55	0.47	0.55	0.47
CPCCS	0.33	0.15	0.48	0.37	0.48	0.37	0.48	0.37

MAGIC shows low replication of homogeneous content because it allows the content to be cached at limited positions. Hence, it produces better performance in terms of redundancy. CPCCS performs better in terms of content redundant caching operations than the other strategies do because of its nature of performing caching partial locations alongside the data routing path. Table 5.4 presents the overall caching performance in numeric values. Hence, it is clear from the results that HPC, CACC, MPC, PDCC, MAGIC, and LeafPopDown showed more redundancy ratio than the proposed CPCCS.

5.3.4 Stretch Ratio

The distance traveled by a consumer Interest toward the content provider is known as stretch [181, 182]. The stretch can be calculated by the following equation:

$$Stretch = \frac{\sum_{i=1}^{I} Hop-traveled}{\sum_{i=1}^{I} Total-Hop} \tag{5.5}$$

where $\sum_{i=1}^{I} Hop - traveled$ shows the number of hops covered by a consumer Interest between the consumer and the content-provider node; $\sum_{i=1}^{|I|} Total - Hop$ represents the total number of hops between the consumer and provider and I illustrates the total number of generated Interests for specific content.

CPCCS caches the popular content close to the consumers at the central position (mutual centrality node), from where all the desired consumers can get their required content. Therefore, it makes the distance between a consumer and a provider smaller and most of the consumer Interests travel through the central position and satisfies from central positions.

Moreover, CPCCS selects diverse popular content to be cached close to the consumers that increase the overall stretch ratio. On the other hand, all the comparing strategies are shown the long stretch path as shown in Figure 5.14, Figure 5.15, Figure 5.16 and Figure 5.17.

WAVE shows the lager path stretch with all cache sizes because it caches the popular content downstream node first that increases the length between consumer and provider.

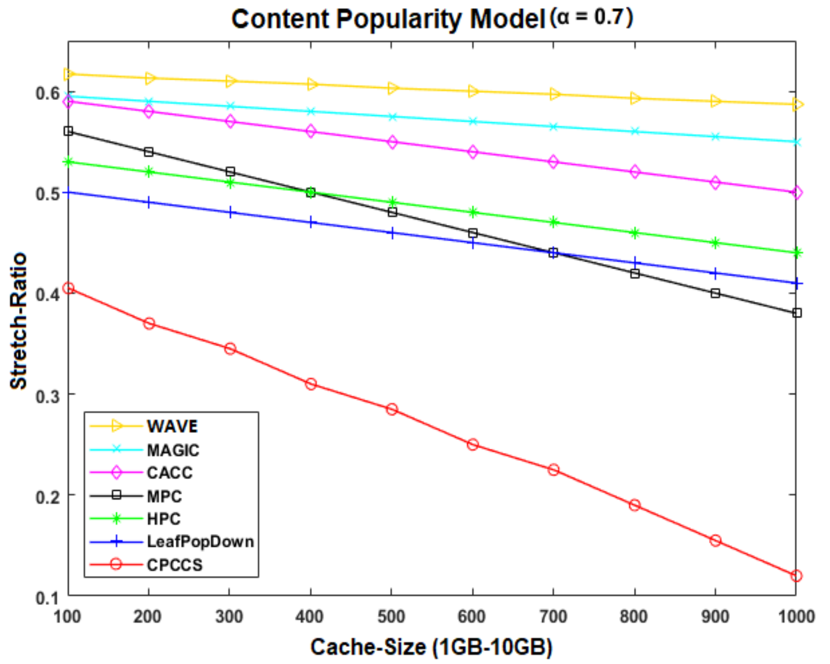

Figure 5.14. Stretch Ratio on Tiger topology (File content)

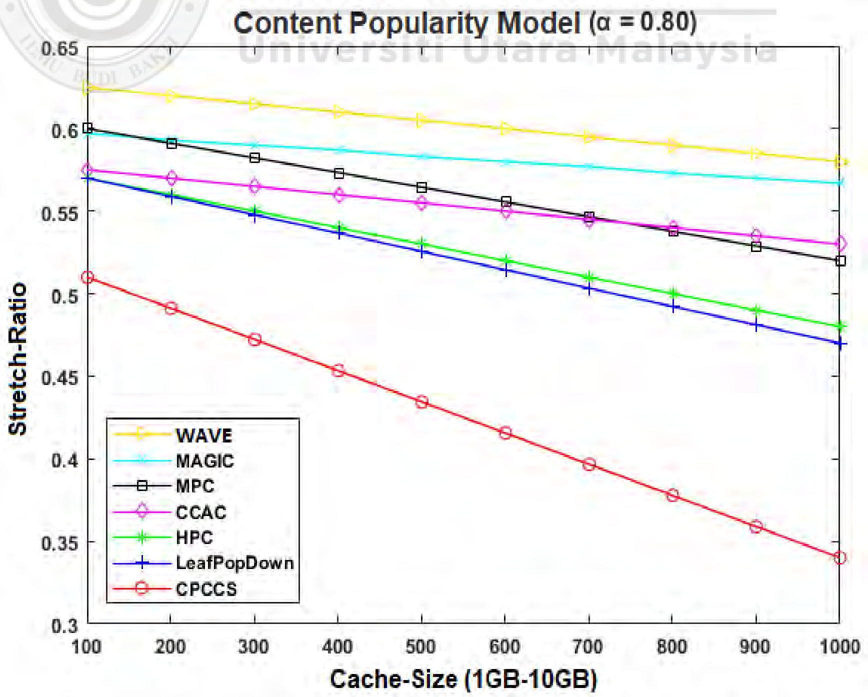

Figure 5.15. Stretch Ratio on Tiger topology (Web content)

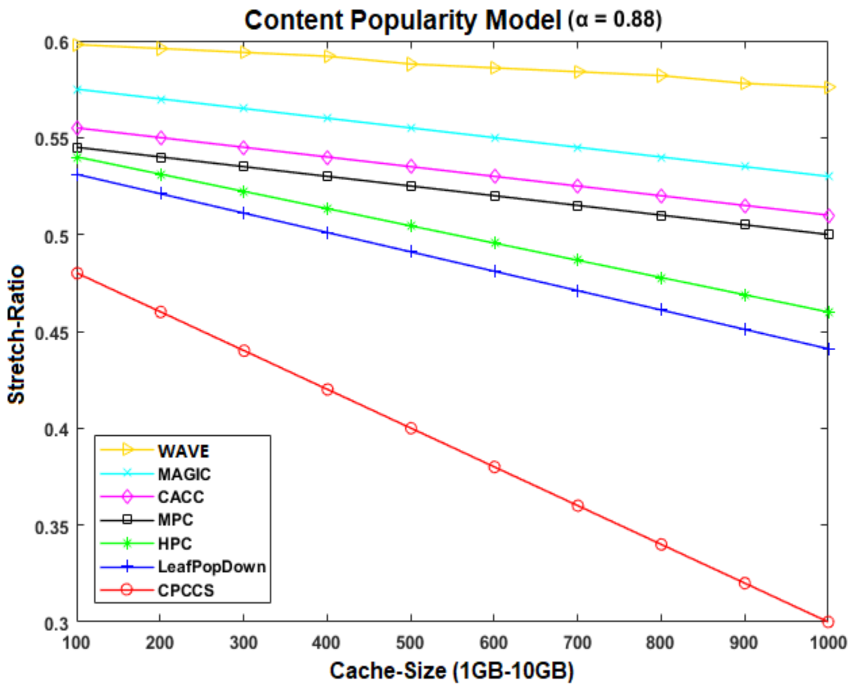

Figure 5.16. Stretch Ratio on Tiger topology (UGC content)

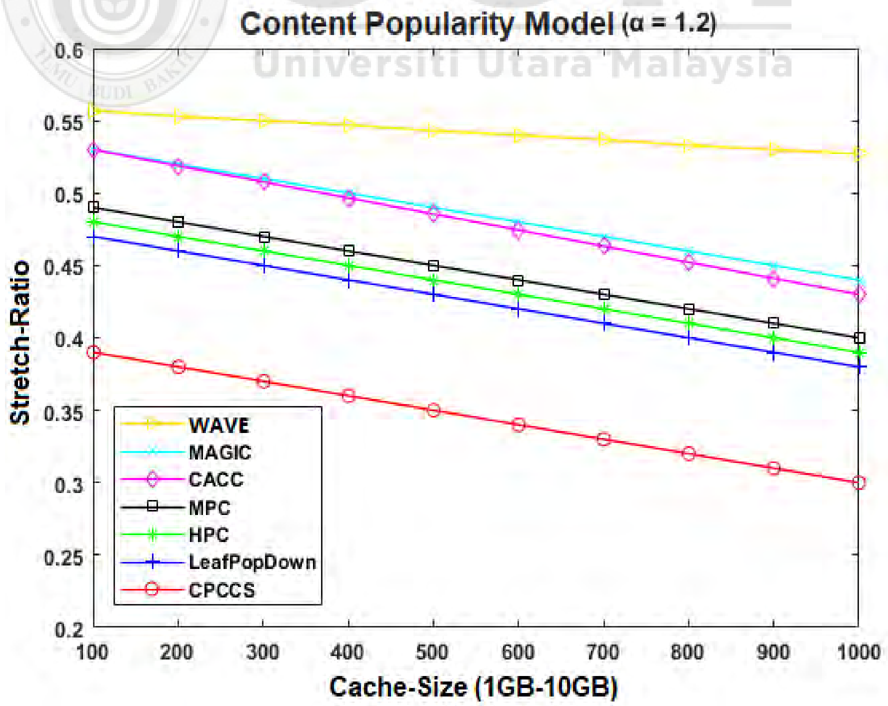

Figure 5.17. Stretch Ratio on Tiger topology (VoD content)

However, WAVE brings content close to the consumer but it takes a long time because of its nature of caching decision. As compared to MAGIC, and CACC, HPC performs better with small and large cache size because it caches a copy of required content at all on-path nodes.

Table 5.5

Stretch numeric results on Tiger

Tiger Topology	**File content ($\alpha = 0.7$)**		**web Content ($\alpha = 0.80$)**		**web Content ($\alpha = 0.88$)**		**web Content ($\alpha = 1.2$)**	
Strategies	1GB	10GB	1GB	10GB	1GB	10GB	1GB	10GB
WAVE	0.62	0.58	0.63	0.58	0.60	0.58	0.56	0.53
MAGIC	0.59	0.56	0.60	0.57	0.57	0.53	0.54	0.44
CCAC	0.58	0.50	0.58	0.53	0.56	0.52	0.54	0.43
MPC	0.56	0.38	0.61	0.52	0.55	0.49	0.49	0.40
HCP	0.53	0.44	0.57	0.48	0.54	0.46	0.48	0.39
LeafPopDown	0.50	0.41	0.57	0.47	0.53	0.44	0.47	0.38
CPCCS	0.41	0.12	0.52	0.34	0.47	0.30	0.39	0.30

Therefore, the subsequent Interests are satisfied with the nearer cached copy of required content. However, MPC and LeafPopDown produce better results in term to reduce the stretch because these strategies cache the required contents close to the consumers. From the results in showing in Table 5.5, it is clear that the proposed caching strategy performs better than all other strategies. CPCCS achieved enhanced results with all cache sizes (1GB to 10GB).

5.4 Summary

In this chapter, the overall performance is elaborated using different topologies and diverse parameter selection. A new caching strategy has been proposed to diminish the multiple replications of similar contents alongside the data routing path. Due to the limited size of the cache, this proposed caching strategy increases the content diversity through caches the heterogeneous contents in the network. Moreover, it reduces the number of redundant contents along the consumer path and decreases the path length in terms of stretch with the selection of a suitable router for content caching. In addition, it enhances the cache hit ratio by the caching of transmitted contents close to the consumers. Therefore, the proposed strategy is evaluated in a simulation environment using SocialCCNSim-Master. The comparison against some state-of-art caching strategies and the consequences show that the proposed CPCCS strategy performs better while the restricted cache size.

CHAPTER SIX
CONCLUSION AND FUTURE WORK

6.1 Introduction

NDN is highly scalable, efficient, and reliable information distribution network architecture. These advantages have encouraged most researchers to modify the current end-to-end sender driven Internet paradigm to a receiver-driven information-centric paradigm. The NDN architectures provide on-path caching (storage for data objects) in the entire network and multicast communication can be produced through data replication. The objective of on-path caching in the NDN architecture is to achieve a scalable, effective, and consistent distribution of information and data objects by using a common communication platform that is available in a dedicated system like a content distribution network. In fact, there is a requirement to modify the present working IP-based network architecture to the NDN architecture. The on-path caching techniques in NDN provide the advantages of disseminating information.

In this study, the problems related to NDN Caching strategies have been evaluated and proposed a new caching strategy named as Compound Popular Content Caching Strategy to resolve the content caching issues while having limited cache capacity. The rest of the chapter is organized as follows: Section 7.2 provides a brief review of overall research, Section 7.3 refers to the research contribution, Section 7.4 describes the research limitations and finally, Section 7.5 gives information about future directions of the study.

6.2 Research summary

Cache management is an important because on-path caching cannot generate efficient results. The management of the cache depends on how the cache is deployed and managed by the network. The effective management of the cache demonstrates a mechanism that places a copy of the transmitted content into the network caching system efficiently. Therefore, in this study proposes a new cache management strategy to enhance the overall performance of NDN-based caching module.

In Chapter One, the research background and introduction about NDN is presented in detailed. Moreover, the research motivation introduced how NDN in a better option for the Internet architecture. In addition, the problems of present NDN cache deployment strategies that arise some critical questions (research questions) and the assumptions (research objectives) to solve these questions is demonstrated in the same chapter. At the end of this chapter, the scope, significance, and overall research plan was described.

Chapter Two is presented the comprehensive introduction about NDN-based on-path caching and its cache deployment mechanisms. On-path caching was divided into two main sections in which the first section explored the probabilistic NDN-based caching strategies while in the second section; the popularity-based caching strategies were explained. At the end of this chapter, the critical analysis of probabilistic and popularity-based caching strategies was elaborated. In Chapter Three, the overall research methodology termed as Design Research Methodology (DRM) is elaborated that is adopted from [133, 183]. In Chapter Four, the proposed cache deployment strategy is expressed in which two mechanisms such as content selection named as Compound Popular Content Section (CPCS) and Compound Popular Content Caching (CPCC) were introduced. Moreover, the chapter also explained the detailed verification and validation.

In Chapter Five, the evaluation of the proposed strategy was presented. Therefore, the proposed strategy is extensively and comparatively studied with other NDN-based caching strategies such as HPC, MPC, LeafPopDown, MAGIC, CCAC, and WAVE through simulations to evaluate the performance in terms of content diversity, cache hit ratio, content redundancy, and stretch ratio. The results show that the proposed caching strategy was performed better than all other strategies.

6.3 Research Contribution

After a study of the state of the art, it is found that diverse caching strategies have been proposed. They include solutions such as leaving copies at every router or detecting the optimal router to create the copy. They use diverse information such as topological structure or the popularity of the content. The first contribution of this research is such that we define a common framework, where we are going to compare some of the common states of the art caching strategies. This common framework is built based on the parameters selected from the literature. The second contribution consists in the comparison of the caching strategies. These will be analyzed according to four metrics: content diversity, content redundancy, cache hit ratio, and stretch. As it is clear from the literature that every caching strategy overcomes other ones in at least one particular scenario, which is detailed in the research. The third and main contribution of this research is to propose a cache deployment strategy for NDN so that it can overcome the limitations of the existing strategies as well as improve the performance of NDN in terms to improve overall cache manageability. Some of the key contributions of this research are given in the following:

1. A comparative analysis of the probabilistic caching strategies to the NDN-based caching literature.
2. A comparative analysis of popularity-based caching strategies to literature related to the NDN-based caching module.
3. Compound Popular Content Selection (CPCS) technique that enhances the threshold to increase the content availability by the selection of least and optimal popular content.
4. Compound Popular Content Caching (CPCC) technique to improve the content manageability through suitable node selection for diverse content caching operation.
5. Coming up with a new algorithm to differentiate regarding interested usage frequency of contents to enhance the content selection mechanism in NDN caching services.
6. Compound Popular Content Caching Strategy (CPCCS) is used to enhance the intermediate position selection by shorten the path stretch and improve the content reachability for subsequent consumers' Interests.
7. Extended the available simulation tool through the implementation of the proposed model for the future comparison.

In this research, a new cache management strategy CPCCS has been proposed to diminish the multiple replications of similar contents alongside the data routing path. The CPCCS is divided into two mechanisms in which CPCS was proposed to select the diverse content to be cached regarding their Interest count while the CPCC mechanism was developed in order to find the optimal location for the caching of OPC and LPC contents. Due to the limited size of the cache, the proposed caching strategy increases the content diversity through caches the heterogeneous contents in the network. Moreover, it reduced the number of redundant contents along the consumer path and decreases the path length in terms of stretch ratio with the selection of a suitable location for content caching. In addition, it enhances the cache hit ratio due to the caching of transmitted content close to the consumers. For the performance evaluation, the proposed strategy is evaluated in a simulation environment after the comparison against some state-of-art caching strategies. Thus, the consequences show that our strategy performs better while the cache size is restricted.

6.4 Research Limitations

In fact, this research has explained the NDN caching-based study using simulation environment through different parameters selection and simulation was established in SocialCCNSim-Master simulator with social network traffic generator (SONETOR [184]). Consequently, the proposed caching strategy was performed better as compared to state-of-art benchmark strategies. However, this study was extended the functionality of CPCCS only for NDN on-path caching technique. Therefore, it is limited to the selected scenario. Meanwhile, the NDN-caching is still in its early stage regarding real-world implementation. Therefore, the proposed scheme could be implemented using the testbed to measure the performance evaluation. With the intense undertaking of discovering NDN testbed systems that support including more unrestricted adaptability, the simulation was entirely prepared using SocialCCNSim-Master.

Moreover, the proposed caching strategy is used to merge sort technique to sort the LPC contents in a list. If the list is too much long it consumes more time to sort. Therefore, this problem can be resolved by sorting the LPC content list without using any sorting algorithm. In addition, the performance of the proposed study was not tested in ad-hoc networks.

6.5 Future Direction

The NDN caching delivers benefits to Internet technologies through the implementation of cache storage. To handle the cache storage, it is most significant to improve the overall caching performance by deploying the efficient cache deployment strategy. Therefore, the proposed caching strategy enhances the caching performance by improving content selection and content caching mechanisms. According to the future prospective, this caching strategy could improve the content dissemination in several technologies such as Internet of Things (IoT), edge cloud computing, blockchain, distributed fog computing, Software Defined Networking (SDN), mobile ad-hoc network (MANET), vehicular ad-hoc network (VANET), and fifth-generation (5G) mobile cellular networks. In regards to genomic data, the extensive use of genomic data sets that are an immense and rapid increase over time is creating problems regarding its transmission. The reason is that these types of data sets are stored at distant (remote) databases and its transmission is supported by IP-based Internet. Therefore, similar genomic data sets are needed to be sent across the remote location several times.

Consequently, several problems are creating regarding the transmission of genomic data sets in which bandwidth, congestion, delay, and high resource consumption. In this situation, NDN provides the number of caching strategies to handle the issues related to genomic data to transmission. Therefore, the present caching strategy can solve these issues in an efficient way by caching the most popular and least popular content near the desired locations. Moreover, it can increase the availability of most popular genomic data sets to respond locally the subsequent requests. Hence, the congestion will be reduced and bandwidth will be used in efficient manner. In addition, the resource (memory and energy) consumption will be decreased.

REFERENCES

[1] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff, "A brief history of the Internet," *ACM SIGCOMM Computer Communication Review,* vol. 39, pp. 22-31, 2009.

[2] Á. O. F. Nascimento, A. A. Prates, and J. F. Lima, "Proposta de um Jogo educacional aplicado ao ensino da arquitetura TCP/IP-EducaRedes."

[3] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou, and D. Dagon, "Detecting Malware Domains at the Upper DNS Hierarchy," in *USENIX security symposium*, 2011, pp. 1-16.

[4] A. L. Lederer, D. J. Maupin, M. P. Sena, and Y. Zhuang, "The technology acceptance model and the World Wide Web," *Decision support systems,* vol. 29, pp. 269-282, 2000.

[5] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. S. Jubert, M. Mazura, M. Harrison, and M. Eisenhauer, "Internet of things strategic research roadmap," *Internet of things-global technological and societal trends,* vol. 1, pp. 9-52, 2011.

[6] A. Merentitis, V. Gazis, E. Patouni, F. Zeiger, M. Huber, N. Frangiadakis, and K. Mathioudakis, "Virtualization as a Driver for the Evolution of the Internet of Things: Remaining Challenges and Opportunities Towards Smart Cities," *International Journal on Advances in Internet Technology Volume 7, Number 1 & 2, 2014,* 2014.

[7] T. Plagemann, R. Canonico, J. Domingo-Pascual, C. Guerrero, and A. Mauthe, "Infrastructures for community networks," in *Content Delivery Networks*, ed: Springer, 2008, pp. 367-388.

[8] J. D. Robinson, P. Muller, T. Noke, T. L. Lim, W. Glausi, and L. Fullerton, "Method of and system for inserting/overlaying markers, data packets and objects relative to viewable content and enabling live social networking, n-dimensional virtual environments and/or other value derivable from the content," ed: Google Patents, 2013.

[9] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, 2009, pp. 1-12.

[10] M. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and B. Mathieu, "A survey of naming and routing in information-centric networks," *IEEE Communications Magazine,* vol. 50, pp. 44-53, 2012.

[11] I. Abdullahi, S. Arif, and S. Hassan, "Survey on caching approaches in Information Centric Networking," *Journal of Network and Computer Applications,* vol. 56, pp. 48-59, 2015.

[12] J. Pan, S. Paul, and R. Jain, "A survey of the research on future internet architectures," *IEEE Communications Magazine,* vol. 49, 2011.

[13] L. Linguaglossa, "Two challenges of Software Networking: Name-based Forwarding and Table Verification," Université Paris Diderot (Paris 7) Sorbonne Paris Cité, 2016.

[14] B. Banerjee, A. Kulkarni, and A. Seetharam, "Greedy Caching: An optimized content placement strategy for information-centric networks," *Computer Networks,* vol. 140, pp. 78-91, 2018.
[15] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. S. Jubert, M. Mazura, M. Harrison, and M. Eisenhauer, "Internet of things strategic research roadmap," *Internet of Things-Global Technological and Societal Trends,* vol. 1, pp. 9-52, 2011.
[16] A. Merentitis, V. Gazis, E. Patouni, F. Zeiger, M. Huber, N. Frangiadakis, and K. Mathioudakis, "Virtualization as a Driver for the Evolution of the Internet of Things: Remaining Challenges and Opportunities Towards Smart Cities," *International Journal on Advances in Internet Technology,* vol. 7, 2014.
[17] M. Garmehi, M. Analoui, M. Pathan, and R. Buyya, "An economic mechanism for request routing and resource allocation in hybrid CDN–P2P networks," *International journal of network management,* vol. 25, pp. 375-393, 2015.
[18] A. X. Liu, C. R. Meiners, and E. Torng, "TCAM Razor: A systematic approach towards minimizing packet classifiers in TCAMs," *IEEE/ACM Transactions on Networking (TON),* vol. 18, pp. 490-500, 2010.
[19] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials,* vol. 16, pp. 1024-1049, 2014.
[20] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine,* vol. 50, 2012.
[21] L. Veltri, G. Morabito, S. Salsano, N. Blefari-Melazzi, and A. Detti, "Supporting information-centric functionality in software defined networks," in *Communications (ICC), 2012 IEEE International Conference on*, 2012, pp. 6645-6650.
[22] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, "Network of Information (NetInf)–An information-centric networking architecture," *Computer Communications,* vol. 36, pp. 721-735, 2013.
[23] A. V. Vasilakos, Z. Li, G. Simon, and W. You, "Information centric network: Research challenges and opportunities," *Journal of Network and Computer Applications,* vol. 52, pp. 1-10, 2015.
[24] D. Grewe, M. Wagner, and H. Frey, "A Domain-Specific Comparison of Information-Centric Networking Architectures for Connected Vehicles," *IEEE Communications Surveys & Tutorials,* 2018.
[25] L. A. Sanchez, "To catch a wave: The Beach Boys and rock historiography," 2012.
[26] T. Sprinkle, *Screw the Valley: A Coast-to-Coast Tour of America s New Tech Startup Culture: New York, Boulder, Austin, Raleigh, Detroit, Las Vegas, Kansas City*: BenBella Books, Inc., 2015.
[27] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A Survey of Green Information-Centric Networking: Research Issues and Challenges," *IEEE Communications Surveys and Tutorials,* vol. 17, pp. 1455-1472, 2015.

[28] I. Abdullahi, S. Arif, and S. Hassan, "Cache-less redundancy using hypergraph in information-centric network," *Advanced Science Letters,* vol. 21, pp. 3546-3549, 2015.

[29] S. Leong, "Navigating the emerging futures in music education," *Journal of Music, Technology & Education,* vol. 4, pp. 233-243, 2012.

[30] Z. Li, Y. Liu, Y. Chen, Y. Xu, and K. Liu, "Performance analysis of a novel 5G architecture via Content-Centric Networking," *Physical Communication,* 2017.

[31] G. Panwar, R. Tourani, T. Mick, A. Mtibaa, and S. Misra, "DICE: Dynamic Multi-RAT Selection in the ICN-enabled Wireless Edge," in *Proceedings of the Workshop on Mobility in the Evolving Internet Architecture*, 2017, pp. 31-36.

[32] M. Alajlan and A. Belghith, "Supporting Seamless Mobility for Real-Time Applications in Named Data Networking," *Procedia Computer Science,* vol. 110, pp. 62-69, 2017.

[33] J. V. Torres, I. D. Alvarenga, R. Boutaba, and O. C. M. Duarte, "An autonomous and efficient controller-based routing scheme for networking Named-Data mobility," *Computer Communications,* vol. 103, pp. 94-103, 2017.

[34] I. Solis, M. Mosko, G. Scott, and A. Walendowski, "CCNx 1.0 Tutorial: Theory and Practice," in *Proceedings of the 2nd International Conference on Information-Centric Networking*, 2015, pp. 1-2.

[35] H. Huang, L. Rui, W. Zheng, D. Niu, X. Qiu, and S. Shao, "General, Practical and Accurate Models for the Performance Analysis of Multi-Cache Systems," *IEICE Transactions on Communications,* vol. 99, pp. 2559-2573, 2016.

[36] Q. N. Nguyen, M. Arifuzzaman, T. Miyamoto, and S. Takuro, "An optimal information centric networking model for the future green network," in *Autonomous Decentralized Systems (ISADS), 2015 IEEE Twelfth International Symposium on*, 2015, pp. 272-277.

[37] K. Katsaros, G. Xylomenos, and G. C. Polyzos, "MultiCache: An overlay architecture for information-centric networking," *Computer Networks,* vol. 55, pp. 936-947, 2011.

[38] I. Ud Din, "Flexpop: A popularity-based caching strategy for multimedia applications in information-centric networking," Universiti Utara Malaysia, 2016.

[39] I. Tosic and P. Frossard, "Dictionary learning for stereo image representation," *IEEE Trans Image Process,* vol. 20, pp. 921-34, Apr 2011.

[40] Cisco, "https://qwilt.com/events/cisco-live-2016/."

[41] J. M. Wein, J. J. Kloninger, M. C. Nottingham, D. R. Karger, and P. A. Lisiecki, "Content delivery network (CDN) content server request handling mechanism with metadata framework support," ed: Google Patents, 2017.

[42] M. Yang, P. Zhu, L. Van Gool, and L. Zhang, "Face recognition based on regularized nearest points between image sets," in *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, 2013, pp. 1-7.

[43] R. Gross and J. Shi, "The cmu motion of body (mobo) database," 2001.

[44] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley, "Face tracking and recognition with visual constraints in real-world videos," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1-8.
[45] "Intel. What happens in an Internet minute, accessed: March 30, 2019. [Online]" *Available:* http://scoop.intel.com/what-happens-in-an-internet-minute/.
[46] X. Wang, B. Wang, X. Bai, W. Liu, and Z. Tu, "Max-margin multiple-instance dictionary learning," in *International Conference on Machine Learning*, 2013, pp. 846-854.
[47] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 13-16.
[48] Z. A. Hua, "Study of ubiquitous learning environment based on Ubiquitous computing," in *Ubi-media Computing (U-Media), 2010 3rd IEEE International Conference on*, 2010, pp. 136-138.
[49] A. Banerjee, X. Chen, J. Erman, V. Gopalakrishnan, S. Lee, and J. Van Der Merwe, "MOCA: a lightweight mobile cloud offloading architecture," in *Proceedings of the eighth ACM international workshop on Mobility in the evolving internet architecture*, 2013, pp. 11-16.
[50] C. Van Neste, R. Hull, T. Abraham, J. Hawk, A. Phani, and T. Thundat, "Wireless single contact power delivery," in *Wireless Power Transfer Conference (WPTC), 2015 IEEE*, 2015, pp. 1-4.
[51] R. Trestian, P. Shah, H. Nguyen, Q.-T. Vien, O. Gemikonakli, and B. Barn, "Towards connecting people, locations and real-world events in a cellular network," *Telematics and Informatics,* vol. 34, pp. 244-271, 2017.
[52] M.-H. Yang, J. Ho, and K.-c. Lee, "Video-based face recognition using probabilistic appearance manifolds," ed: Google Patents, 2009.
[53] M. Ain, D. Trossen, P. Nikander, S. Tarkoma, K. Visala, K. Rimey, T. Burbridge, J. Rajahalme, J. Tuononen, and P. Jokela, "D2. 3–architecture definition, component descriptions, and requirements," *Deliverable, PSIRP 7th FP EU-funded project,* 2009.
[54] G. Piro, S. Signorello, M. R. Palattella, L. A. Grieco, G. Boggia, and T. Engel, "Understanding the social impact of ICN: between myth and reality," *AI & SOCIETY,* vol. 32, pp. 401-419, 2017.
[55] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache "less for more" in information-centric networks (extended version)," *Computer Communications,* vol. 36, pp. 758-770, 2013.
[56] H. Jin, D. Xu, C. Zhao, and D. Liang, "Information-centric mobile caching network frameworks and caching optimization: a survey," *EURASIP Journal on Wireless Communications and Networking,* vol. 2017, p. 33, 2017.
[57] M. D'Ambrosio, "D-6.2 second NetInf architecture description," *The Network of the Future Project,* vol. 216041, 2010.
[58] P. Si, H. Yue, Y. Zhang, and Y. Fang, "Spectrum management for proactive video caching in information-centric cognitive radio networks," *IEEE Journal on Selected Areas in Communications,* vol. 34, pp. 2247-2259, 2016.

[59] G. Zhang, Y. Li, and T. Lin, "Caching in information centric networking: A survey," *Computer Networks,* vol. 57, pp. 3128-3141, 2013.
[60] M. A. Naeem, S. A. Nor, S. Hassan, and B.-S. Kim, "Performances of Probabilistic Caching Strategies in Content Centric Networking," *IEEE Access,* 2018.
[61] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, and C. Papadopoulos, "Named data networking (ndn) project," *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC,* vol. 157, p. 158, 2010.
[62] E. G. AbdAllah, M. Zulkernine, and H. S. Hassanein, "Detection and prevention of malicious requests in ICN routing and caching," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015, pp. 1741-1748.
[63] C. Bernardini, T. Silverston, and O. Festor, "Mpc: Popularity-based caching strategy for content centric networks," in *Communications (ICC), 2013 IEEE International Conference on*, 2013, pp. 3619-3623.
[64] C. Ghali, G. Tsudik, and C. A. Wood, "When encryption is not enough: privacy attacks in content-centric networking," in *Proceedings of the 4th ACM Conference on Information-Centric Networking*, 2017, pp. 1-10.
[65] Q. Chen, R. Xie, F. R. Yu, J. Liu, T. Huang, and Y. Liu, "Transport control strategies in named data networking: A survey," *IEEE Communications Surveys & Tutorials,* vol. 18, pp. 2052-2083, 2016.
[66] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Communications Surveys & Tutorials,* vol. 17, pp. 1473-1499, 2015.
[67] A. Ioannou and S. Weber, "A survey of caching policies and forwarding mechanisms in information-centric networking," *IEEE Communications Surveys & Tutorials,* vol. 18, pp. 2847-2886, 2016.
[68] H. Khattak, N. Ul Amin, I. U. Din, Insafullah, and J. Iqbal, "LeafPopDown: Leaf Popular Down Caching Strategy for Information-Centric Networking," *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS,* vol. 9, pp. 148-151, 2018.
[69] S. Arshad, M. A. Azam, M. H. Rehmani, and J. Loo, "Information-Centric Networking based Caching and Naming Schemes for Internet of Things: A Survey and Future Research Directions," *arXiv preprint arXiv:1710.03473,* 2017.
[70] S. Rahel, A. Jamali, and S. El Kafhali, "Energy-efficient on caching in named data networking: A survey," in *Cloud Computing Technologies and Applications (CloudTech), 2017 3rd International Conference of*, 2017, pp. 1-8.
[71] Y. Wang, M. Xu, and Z. Feng, "Hop-based probabilistic caching for information-centric networks," in *2013 IEEE Global Communications Conference (GLOBECOM)*, 2013, pp. 2102-2107.

[72] C. Bernardini, T. Silverston, and O. Festor, "MPC: Popularity-based caching strategy for content centric networks," in *2013 IEEE international conference on communications (ICC)*, 2013, pp. 3619-3623.

[73] J. Ren, W. Qi, C. Westphal, J. Wang, K. Lu, S. Liu, and S. Wang, "Magic: A distributed max-gain in-network caching strategy in information-centric networks," in *2014 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*, 2014, pp. 470-475.

[74] S.-W. Lee, D. Kim, Y.-B. Ko, J.-H. Kim, and M.-W. Jang, "Cache capacity-aware CCN: Selective caching and cache-aware routing," in *2013 IEEE Global Communications Conference (GLOBECOM)*, 2013, pp. 2114-2119.

[75] K. Cho, M. Lee, K. Park, T. T. Kwon, Y. Choi, and S. Pack, "Wave: Popularity-based and collaborative in-network caching for content-oriented networks," in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, 2012, pp. 316-321.

[76] W. D. Abdullah, A. M. MonzerHabbal, and M. B. Mahmuddin, "Evaluation of user behavior and network performance in Malaysian Institution of Higher Education (MIHE) of wireless network," in *New Trends in Information & Communications Technology Applications (NTICT), 2017 Annual Conference on*, 2017, pp. 46-51.

[77] I. U. Din, S. Hassan, M. K. Khan, M. Guizani, O. Ghazali, and A. Habbal, "Caching in Information-Centric Networking: Strategies, Challenges, and Future Research Directions," *IEEE Communications Surveys & Tutorials,* 2017.

[78] M. A. Naeem and S. A. Nor, "A survey of content placement strategies for content-centric networking," in *AIP Conference Proceedings*, 2016, p. 020078.

[79] V. Sourlas, L. Gkatzikis, P. Flegkas, and L. Tassiulas, "Distributed cache management in information-centric networks," *IEEE Transactions on Network and Service Management,* vol. 10, pp. 286-299, 2013.

[80] J. Li, H. Wu, B. Liu, J. Lu, Y. Wang, X. Wang, Y. Zhang, and L. Dong, "Popularity-driven coordinated caching in named data networking," in *Proceedings of the eighth ACM/IEEE symposium on Architectures for networking and communications systems*, 2012, pp. 15-26.

[81] A. Papa, G. Santoro, L. Tirabeni, and F. Monge, "Social media as tool for facilitating knowledge creation and innovation in small and medium enterprises," *Baltic Journal of Management,* 2018.

[82] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *ACM SIGCOMM Computer Communication Review,* vol. 44, pp. 66-73, 2014.

[83] M. Conti, P. Gasti, and M. Teoli, "A lightweight mechanism for detection of cache pollution attacks in named data networking," *Computer Networks,* vol. 57, pp. 3178-3191, 2013.

[84] L. Yao, Z. Fan, J. Deng, X. Fan, and G. Wu, "Detection and Defense of Cache Pollution Attacks Using Clustering in Named Data Networks," *IEEE Transactions on Dependable and Secure Computing,* 2018.

[85] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "An energy-efficient distributed in-network caching scheme for green content-centric networks," *Computer Networks,* vol. 78, pp. 119-129, 2015.

[86] Y. Qin, W. Yang, and W. Liu, "A Probability-based Caching Strategy with Consistent Hash in Named Data Networking," in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*, 2018, pp. 67-72.

[87] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A case for stateful forwarding plane," *Computer Communications,* vol. 36, pp. 779-791, 2013.

[88] M. Lee, J. Song, K. Cho, S. Pack, J. Kangasharju, and Y. Choi, "Content discovery for information-centric networking," *Computer Networks,* vol. 83, pp. 1-14, 2015.

[89] E. G. AbdAllah, M. Zulkernine, and H. S. Hassanein, "DADI: Defending against distributed denial of service in information-centric networking routing and caching," *Security and Privacy,* p. e16, 2018.

[90] A. Araldo, D. Rossi, and F. Martignon, "Cost-aware caching: Caching more (costly items) for less (ISPs operational expenditures)," *IEEE Transactions on Parallel and Distributed Systems,* vol. 27, pp. 1316-1330, 2016.

[91] A. Ioannou and S. Weber, "Towards on-path caching alternatives in information-centric networks," in *Local Computer Networks (LCN), 2014 IEEE 39th Conference on*, 2014, pp. 362-365.

[92] A. Seetharam, "On Caching and Routing in Information-Centric Networks," *IEEE Communications Magazine,* vol. 56, pp. 204-209, 2018.

[93] L. Rui, H. Guo, H. Huang, R. Shi, and X. Qiu, "MUPF: Multiple Unicast Path Forwarding in Content-Centric VANETs," *Ad Hoc Networks,* 2018.

[94] R. Alubady, M. Salman, S. Hassan, and A. Habbal, "Review of Name Resolution and Data Routing for Information Centric Networking," in *The 4th International Conference on Internet Applications, Protocols and Services (NETAPPS2015)*, 2015, pp. 48-53.

[95] A. M. Alberti, M. A. F. Casaroli, D. Singh, and R. da Rosa Righi, "Naming and name resolution in the future internet: Introducing the NovaGenesis approach," *Future Generation Computer Systems,* vol. 67, pp. 163-179, 2017.

[96] M. Ambrosin, A. Compagno, M. Conti, C. Ghali, and G. Tsudik, "Security and Privacy Analysis of National Science Foundation Future Internet Architectures," *IEEE Communications Surveys & Tutorials,* vol. 20, pp. 1418-1442, 2018.

[97] S. Arianfar, P. Nikander, and J. Ott, "On content-centric router design and implications," in *Proceedings of the Re-Architecting the Internet Workshop*, 2010, p. 5.

[98] I. Psaras, W. K. Chai, and G. Pavlou, "In-network cache management and resource allocation for information-centric networks," *IEEE Transactions on Parallel and Distributed Systems,* vol. 25, pp. 2920-2931, 2014.

[99] O. Ascigil, V. Sourlas, I. Psaras, and G. Pavlou, "A native content discovery mechanism for the information-centric networks," in *Proceedings of the 4th ACM Conference on Information-Centric Networking*, 2017, pp. 145-155.

[100] A. Banerjee, B. Banerjee, A. Seetharam, and C. Tellambura, "Content search and routing under custodian unavailability in information-centric networks," *Computer Networks,* vol. 141, pp. 92-101, 2018.
[101] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, and K. Drira, "Named Data Networking: A promising architecture for the Internet of things (IoT)," 2017.
[102] C. Bernardini, "Stratégies de Cache basées sur la popularité pour Content Centric Networking," Université de Lorraine, 2015.
[103] C. Bernardini, T. Silverston, and O. Festor, "A comparison of caching strategies for content centric networking," in *Global Communications Conference (GLOBECOM), 2015 IEEE*, 2015, pp. 1-6.
[104] C. Y. Chang and M. S. Chang, "A hybrid coordination approach of in-network caching for named data networking," *International Journal of Future Generation Communication and Networking,* vol. 9, pp. 285-300, 2016.
[105] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache “less for more” in information-centric networks," in *International Conference on Research in Networking*, 2012, pp. 27-40.
[106] L. Ding, J. Wang, Y. Sheng, and L. Wang, "AN EFFICIENT RANKING-MATCHED CACHING STRATEGY FOR INFORMATION CENTRIC NETWORKING," *INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING INFORMATION AND CONTROL,* vol. 14, pp. 519-535, 2018.
[107] M. Dräxler and H. Karl, "Efficiency of on-path and off-path caching strategies in information centric networks," in *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*, 2012, pp. 581-587.
[108] S. Eum, K. Nakauchi, M. Murata, Y. Shoji, and N. Nishinaga, "CATT: potential based routing with content caching for ICN," in *Proceedings of the second edition of the ICN workshop on Information-centric networking*, 2012, pp. 49-54.
[109] C. Fang, H. Yao, Z. Wang, W. Wu, X. Jin, and F. R. Yu, "A survey of mobile information-centric networking: Research issues and challenges," *IEEE Communications Surveys & Tutorials,* 2018.
[110] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Scalable mobile backhauling via information-centric networking," in *Local and Metropolitan Area Networks (LANMAN), 2015 IEEE International Workshop on*, 2015, pp. 1-6.
[111] C. Cisco, "Cisco visual networking index: global mobile data traffic forecast, 2016–2021," *Cisco white paper,* 2017.
[112] X.-d. Cui, T. Huang, L. Jiang, L. Li, J.-y. Chen, and Y.-j. LIU, "Design of in-network caching scheme in CCN based on grey relational analysis," *The Journal of China Universities of Posts and Telecommunications,* vol. 21, pp. 1-8, 2014.
[113] A. Dabirmoghaddam, M. M. Barijough, and J. Garcia-Luna-Aceves, "Understanding optimal caching and opportunistic caching at the edge of information-centric networks," in *Proceedings of the 1st ACM conference on information-centric networking*, 2014, pp. 47-56.

[114] V. Pacifici and G. Dán, "Paper D: Distributed Algorithms for Content Allocation in Inter-connected Content Distribution Networks," *Resource Allocation in Operator-owned Content Delivery Systems,* p. 29, 2016.

[115] S. Puzhavakath Narayanan, Y. S. Nam, A. Sivakumar, B. Chandrasekaran, B. Maggs, and S. Rao, "Reducing latency through page-aware management of web objects by content delivery networks," in *ACM SIGMETRICS Performance Evaluation Review*, 2016, pp. 89-100.

[116] G. Rossini and D. Rossi, "Evaluating CCN multi-path interest forwarding strategies," *Computer Communications,* vol. 36, pp. 771-778, 2013.

[117] H. Nakajima, T. Tsutsui, H. Urabayashi, M. Yamamoto, E. Rosensweig, and J. F. Kurose, "Performance Evaluation of Partial Deployment of an In-Network Cache Location Guide Scheme, Breadcrumbs," *IEICE Transactions on Communications,* vol. 99, pp. 157-166, 2016.

[118] Y. Li, T. Zhang, X. Xu, Z. Zeng, and Y. Liu, "Content popularity and node level matched based probability caching for content centric networks," in *Communications in China (ICCC), 2016 IEEE/CIC International Conference on*, 2016, pp. 1-6.

[119] R. S. Abbas, "Performance evaluation of caching placement algorithms in named data network for video on demand service," Universiti Utara Malaysia, 2016.

[120] H. Wu, J. Li, J. Zhi, Y. Ren, and L. Li, "Design and Evaluation of Probabilistic Caching in Information-Centric Networking," *IEEE Access,* 2018.

[121] W. Ding, Z. Yan, and R. H. Deng, "A survey on future Internet security architectures," *IEEE Access,* vol. 4, pp. 4374-4393, 2016.

[122] G. Carofiglio, L. Mekinda, and L. Muscariello, "LAC: Introducing latency-aware caching in Information-Centric Networks," in *Local Computer Networks (LCN), 2015 IEEE 40th Conference on*, 2015, pp. 422-425.

[123] J. Du, Y. He, W. Da, and J. Xiao, "Provisioning a cluster of distributed computing platform based on placement strategy," ed: Google Patents, 2016.

[124] N. Abani, T. Braun, and M. Gerla, "Proactive caching with mobility prediction under uncertainty in information-centric networks," in *Proceedings of the 4th ACM Conference on Information-Centric Networking*, 2017, pp. 88-97.

[125] Y. Zhu, Z.-K. Mi, and W.-N. Wang, "A Probability Caching Decision Policy with Evicted Copy Up in Content Centric Networking," *Journal of Internet Technology,* vol. 99, pp. 1-11, 2015.

[126] M. Namba, "A recursive estimation of network state for improving probabilistic caching," in *Soft Computing and Pattern Recognition (SoCPaR), 2015 7th International Conference of*, 2015, pp. 270-273.

[127] W. Li, "Popularity-driven caching strategy for dynamic adaptive streaming over information-centric networks," 2015.

[128] H. Wu, J. Li, T. Pan, and B. Liu, "A novel caching scheme for the backbone of named data networking," in *Communications (ICC), 2013 IEEE International Conference on*, 2013, pp. 3634-3638.

[129] Y. Sun, T. Zhang, R. Wang, W. Feng, and P. Chen, "Topology Potential Based Probability Caching Strategy for Content Centric Ad Hoc Networks," *Journal of Residuals Science & Technology,* vol. 13, 2016.
[130] H. Wu, J. Li, and J. Zhi, "MBP: A max-benefit probability-based caching strategy in information-centric networking," in *Communications (ICC), 2015 IEEE International Conference on*, 2015, pp. 5646-5651.
[131] H. Tian, Y. Otsuka, M. Mohri, Y. Shiraishi, and M. Morii, "Leveraging In-Network Caching in Vehicular Network for Content Distribution," *International Journal of Distributed Sensor Networks,* vol. 12, p. 8972950, 2016.
[132] C. Bernardini, T. Silverston, and F. Olivier, "Towards popularity-based caching in content centric networks," in *RESCOM 2012*, 2012.
[133] L. T. Blessing and A. Chakrabarti, *DRM: A Design Reseach Methodology*: Springer, 2009.
[134] F. J. Suárez, P. Nuño, J. C. Granda, and D. F. García, "Computer networks performance modeling and simulation," in *Modeling and Simulation of Computer Networks and Systems*, ed: Elsevier, 2015, pp. 187-223.
[135] A. Chakrabarti and L. Blessing, "A review of theories and models of design," *Journal of the Indian Institute of Science,* vol. 95, pp. 325-340, 2016.
[136] G. A. Wainer, *Discrete-event modeling and simulation: a practitioner's approach*: CRC press, 2017.
[137] J. L. Hicks, T. K. Uchida, A. Seth, A. Rajagopal, and S. L. Delp, "Is my model good enough? Best practices for verification and validation of musculoskeletal models and simulations of movement," *Journal of biomechanical engineering,* vol. 137, 2015.
[138] L. Gulyás and G. Kampis, "Models, Representations and Comparisons in Computer Simulations," *Procedia Computer Science,* vol. 66, pp. 5-12, 2015.
[139] R. Bagrodia, K. Tang, and J. Hsu, "Method and system for enhancing performance of a physical network under real-time control using simulation of a reference model," ed: Google Patents, 2010.
[140] L. Saino, I. Psaras, and G. Pavlou, "Icarus: a caching simulator for information centric networking (icn)," in *SimuTools*, 2014, pp. 66-75.
[141] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," *University of California, Los Angeles, Tech. Rep,* vol. 4, 2012.
[142] M. Tortelli, D. Rossi, G. Boggia, and L. A. Grieco, "ICN software tools: survey and cross-comparison," *Simulation Modelling Practice and Theory,* vol. 63, pp. 23-46, 2016.
[143] M. Hassan and R. Jain, *High performance TCP/IP networking* vol. 29: Prentice Hall Upper Saddle River, NJ, 2003.
[144] M. M. F. Hamdi, A. Habbal, N. H. Zakaria, and S. Hassan, "Evaluation of caching strategies in content-centric networking (CCN) for mobile and social networking environment," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC),* vol. 10, pp. 1-6, 2018.
[145] I. U. Din, S. Hassan, M. K. Khan, M. Guizani, O. Ghazali, and A. Habbal, "Caching in information-centric networking: Strategies, challenges, and future research directions," *IEEE Communications Surveys & Tutorials,* vol. 20, pp. 1443-1474, 2017.

[146] I. Abdullahi, S. Arif, and M. Omar, "Positioning Cache in Information-Centric Networking," *Advanced Science Letters,* vol. 23, pp. 5346-5351, 2017.
[147] C. Bernardini, T. Silverston, and O. Festor, "A comparison of caching strategies for content centric networking," in *2015 IEEE Global Communications Conference (GLOBECOM)*, 2015, pp. 1-6.
[148] C. Yanqing, W. Muqing, Z. Min, and W. Kaili, "Socially-aware NodeRank-based caching strategy for Content-Centric Networking," in *2016 International Symposium on Wireless Communication Systems (ISWCS)*, 2016, pp. 297-303.
[149] B. Ohlman, E. Davies, S. Spirou, K. Pentikousis, and G. Boggia, "Information-Centric Networking: Evaluation and Security Considerations," 2016.
[150] L. Wang, S. Bayhan, J. Ott, J. Kangasharju, and J. Crowcroft, "Understanding Scoped-Flooding for Content Discovery and Caching in Content Networks," *IEEE Journal on Selected Areas in Communications,* 2018.
[151] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning* vol. 1: MIT press Cambridge, 2016.
[152] R. Diestel, *Graph theory*: Springer Publishing Company, Incorporated, 2018.
[153] L. R. Foulds, *Graph theory applications*: Springer Science & Business Media, 2012.
[154] S. Shirinivas, S. Vetrivel, and N. Elango, "Applications of graph theory in computer science an overview," *International Journal of Engineering Science and Technology,* vol. 2, pp. 4610-4621, 2010.
[155] N. Somu, M. G. Raman, K. Kirthivasan, and V. S. Sriram, "Hypergraph based feature selection technique for medical diagnosis," *Journal of medical systems,* vol. 40, p. 239, 2016.
[156] M. Naeem, R. Ali, B.-S. Kim, S. Nor, and S. Hassan, "A periodic caching strategy solution for the smart city in information-centric Internet of Things," *Sustainability,* vol. 10, p. 2576, 2018.
[157] I. U. Din, S. Hassan, and A. Habbal, "SocialCCNSim: A simulator for caching strategies in information-centric networking," *Advanced Science Letters,* vol. 21, pp. 3505-3509, 2015.
[158] J. Ren, W. Qi, C. Westphal, J. Wang, K. Lu, S. Liu, and S. Wang, "Magic: A distributed max-gain in-network caching strategy in information-centric networks," in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, 2014, pp. 470-475.
[159] Y. Xu, Z. Wang, Y. Li, F. Chen, T. Lin, and W. Niu, "Request routing through collaborative in-network caching for bandwidth optimization: a methodology," *Transactions on Emerging Telecommunications Technologies,* vol. 28, 2017.
[160] P. K. Sharma, S. Rathore, and J. H. Park, "Multilevel learning based modeling for link prediction and users' consumption preference in Online Social Networks," *Future Generation Computer Systems,* 2017.
[161] X. Gabaix, "Zipf's law for cities: an explanation," *The Quarterly journal of economics,* vol. 114, pp. 739-767, 1999.

[162] L. A. Adamic and B. A. Huberman, "Zipf's law and the Internet," *Glottometrics,* vol. 3, pp. 143-150, 2002.
[163] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *Proceedings of the second edition of the ICN workshop on Information-centric networking*, 2012, pp. 55-60.
[164] D. Rossi and G. Rossini, "Caching performance of content centric networks under multi-path routing (and more)," *Relatório técnico, Telecom ParisTech,* pp. 1-6, 2011.
[165] J. Khan, C. Westphal, J. Garcia-Luna-Aceves, and Y. Ghamri-Doudane, "NICE: Network-oriented Information-centric Centrality for Efficiency in Cache Management," in *ACM Conference on Information-Centric Networking (ACM-ICN 2018)*, 2018.
[166] Y. Carlinet, B. Kauffmann, P. Olivier, and A. Simonian, "Trace-based analysis for caching multimedia services," *Orange Labs Technical Report,* 2011.
[167] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, "Understanding user behavior in large-scale video-on-demand systems," in *ACM SIGOPS Operating Systems Review*, 2006, pp. 333-344.
[168] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, "Impact of traffic mix on caching performance in a content-centric network," in *2012 Proceedings IEEE INFOCOM Workshops*, 2012, pp. 310-315.
[169] M. Bilal and S.-G. Kang, "Time aware least recent used (TLRU) cache management policy in ICN," in *16th International Conference on Advanced Communication Technology*, 2014, pp. 528-532.
[170] L. Saino, I. Psaras, and G. Pavlou, "Icarus: a caching simulator for information centric networking (ICN)," in *Proceedings of the 7th International ICST conference on Simulation Tools and Techniques*, 2014, pp. 66-75.
[171] A. Lareida, R. Pernischova, B. B. Rodrigues, and B. Stiller, "Abstracting. torrent content consumption into two-mode graphs and their projection to content networks (ConNet)," in *Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on*, 2017, pp. 151-159.
[172] R. Jmal and L. C. Fourati, "An OpenFlow architecture for managing content-centric-network (OFAM-CCN) based on popularity caching strategy," *Computer Standards & Interfaces,* vol. 51, pp. 22-29, 2017.
[173] S. C. Steely Jr, W. C. Hasenplaugh, J. S. Emer, and S. Subramaniam, "High bandwidth full-block write commands," ed: Google Patents, 2018.
[174] J. Niehues, "Non-uniform sampling in bandwidth constrained data acquisition systems," ed: Google Patents, 2018.
[175] P. Back, S. Zeytinoglu, A. Ijaz, M. Kroner, and A. Imamoğlu, "Realization of an electrically tunable narrow-bandwidth atomically thin mirror using monolayer MoSe 2," *Physical review letters,* vol. 120, p. 037401, 2018.
[176] İ. Gülçın, M. Oktay, E. Kıreçcı, and Ö. İ. Küfrevıoğlu, "Screening of antioxidant and antimicrobial activities of anise (Pimpinella anisum L.) seed extracts," *Food chemistry,* vol. 83, pp. 371-382, 2003.

[177] Z. Chen, N. Pappas, and M. Kountouris, "Probabilistic caching in wireless D2D networks: Cache hit optimal versus throughput optimal," *IEEE Communications Letters,* vol. 21, pp. 584-587, 2017.
[178] Naeem et al, "Information Centric Progressive Content Placement Cache for Multimedia data contents " *Journal of Advanced Research in Dynamical and Control Systems* vol. 10, pp. 1291-1301, 2018.
[179] M. J. Joyner and J. A. Dempsey, "Physiological Redundancy and the Integrative Responses to Exercise," *Cold Spring Harbor perspectives in medicine,* vol. 8, p. a029660, 2018.
[180] J. Tian, C. Mao, D. Wang, S. Nie, and Y. Yang, "A Short-time Transition and Cost Saving Redundancy Scheme for Medium-voltage Three-phase Cascaded H-bridge Electronic Power Transformer," *IEEE Transactions on Power Electronics,* 2018.
[181] C. W. Badenhop, S. R. Graham, B. W. Ramsey, B. E. Mullins, and L. O. Mailloux, "The Z-Wave routing protocol and its security implications," *Computers & Security,* vol. 68, pp. 112-129, 2017.
[182] J. A. F. F. Dias, "Performance of management solutions and cooperation approaches for vehicular delay-tolerant networks," 2017.
[183] A. Habbal, "TCP Sintok: Transmission control protocol with delay-based loss detection and contention avoidance mechanisms for mobile ad hoc networks," Universiti Utara Malaysia, 2014.
[184] C. Bernardini, T. Silverston, and O. Festor, "SONETOR: A social network traffic generator," in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 3734-3739.