PERFORMANCE EVALUATION OF CACHING PLACEMENT ALGORITHMS IN NAMED DATA NETWORK FOR VIDEO ON DEMAND SERVICE

RASHA SALEEM ABBAS

MASTER OF SCIENCE (INFORMATION TECHNOLOGY)
UNIVERSITI UTARA MALAYSIA
2016

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

Tujuan kajian ini adalah untuk menilai prestasi algoritma penempatan caching (LCD, LCE, Prob, Pprob, Cross, Centrality, dan Rand) dalam 'Named Data Network' (NDN) untuk 'Video-on-Demand' (VoD) untuk meningkatkan kualiti dan akses kelewatan perkhidmatan yang disebabkanb oleh kekerapan muat turun yang rendah. Tambahan pula, masalah trafik video berat melambatkan prestasi VoD dalam kes skala besar 'Content- Centric Networks' (CCN). Dua peringkat aktiviti yang mengakibatkan hasil kajian: Yang pertama adalah dengan memeriksa aktiviti penyelidikan eksperimen untuk menentukan punca prestasi kelewatan dalam algoritma cache NDN yang digunakan dalam beban kerja VoD. Aktiviti kedua ialah pelaksanaan tujuh algoritma penempatan cache pada kandungan 'CloudTV' dari segi metrik prestasi utama (masa tunda, nisbah hit purata, jumlah pengurangan jejak rangkaian, dan pengurangan beban). Simulator NS3 dan topologi Internet digunakan untuk menilai dan menganalisis hasil setiap algoritma, dan untuk membandingkan keputusan berdasarkan saiz cache (1GB, 10GB, 100GB, 1TB). Oleh itu, kajian ini membuktikan bahawa pertamanya, sebab utama kelewatan disebabkan oleh lalu lintas video dengan permintaan pengguna yang berbeza. Selain peningkatan pesat dalam permintaan pengguna untuk video dalam talian, kapasiti simpanan juga akan meningkat dan seterusnya membuat replikasi data penyimpanan keseluruhan yang hampir tidak kelihatan. Kedua, hasil kajian membuktikan bahawa peningkatan kapasiti cache menyebabkan rangsangan ketara dalam nisbah purata hit, pengurangan dalam beban pelayan, dan pengurangan dalam jejak rangkaian, yang mengakibatkan mengurangkan masa tunda. Ketiga, berdasarkan keputusan yang diperolehi, didapati bahawa kepusatan secara algoritma penempatan cache tidak memuaskan, kerana ia menghasilkan nilai yang paling teruk dalam purata nisbah hit cache dan dalam jumlah pengurangan jejak rangkaian. Di samping itu, untuk video dalam talian, kapasiti simpanan juga akan meningkat dan seterusnya membuat replikasi data penyimpanan keseluruhan yang hampir tidak dapat dikesan. Selain itu, maklum balas yang berterusan kepada permintaan video pengguna dalam talian meningkatkan trafik video dan prestasi perkhidmatan VoD yang dipaparkan serta menjejaskan kandungan caching dalam router.

Kata kunci: Caching Placement Algorithms, Named Data Network (NDN), Video-on- Demand (VoD), Content-Centric Networks (CCN).

Abstract

The purpose of this study is to evaluate the performance of caching placement algorithms (LCD, LCE, Prob, Pprob, Cross, Centrality, and Rand) in Named Data Network (NDN) for Video on Demand (VoD). This study aims to increment the service quality and to decrement the time of download. There are two stages of activities resulted in the outcome of the study: The first is to determine the causes of delay performance in NDN cache algorithms used in VoD workload. The second activity is the evaluation of the seven cache placement algorithms on the cloud of video content in terms of the key performance metrics: delay time, average cache hit ratio, total reduction in the network footprint, and reduction in load. The NS3 simulations and the Internet2 topology were used to evaluate and analyze the findings of each algorithm, and to compare the results based on cache sizes: 1GB, 10GB, 100GB, and 1TB. This study proves that the different user requests of online videos would lead to delay in network performance. In addition to that the delay also caused by the high increment of video requests. Also, the outcomes led to conclude that the increase in cache capacity leads to make the placement algorithms have a significant increase in the average cache hit ratio, a reduction in server load, and the total reduction in network footprint, which resulted in obtaining a minimized delay time. In addition to that, a conclusion was made that Centrality is the worst cache placement algorithm based on the results obtained.

Keywords: Caching Placement Algorithms, Named Data Network (NDN), Video on Demand (VoD), Content Centric Networks (CCN).

Acknowledgements

In the name of Allah the Merciful Allah is the Light of the heavens and the earth. The example of His light is like a niche within which is a lamp, the lamp is within glass, the glass as if it were a pearly [white] star lit from [the oil of] a blessed olive tree, neither of the east nor of the west, whose oil would almost glow even if untouched by fire. Light upon light. Allah guides to His light whom He wills. And Allah presents examples for the people, and Allah is Knowing of all things. Surat Al-Nur / A-35

Firstly, for Allah, Alhamdulillah. My deepest gratitude is consecrated to my supervisors: Dr. Ahmad Suki Che Mohamed Arif and Dr. Adib Habbal, for their continuous guidance, fruitful feedback, moral support, and sharing of all their research experiences throughout these challenging years. They have eagerly provided a surplus of advices and constructive comments as well as optimism and encouragement at times when things were not looking sunny. Their detailed and constructive comments have helped me to better shape my research ideas.

Besides them, my gratitude to all my colleagues (School of Computing, Universiti Utara Malaysia) in the Master journey, among them is Professor Dr. Suhaidi Hassan, Dr. Mohd. Hasbullah Omar, Dr. Shahrudin Awang Nor, Dr. Mohammed K.M. Madi, Dr. Norliza Katuk, and Dr. Rohaida Romli, Dr. Massudi Mahmuddin, Dr. Nur Haryani Zakaria, and many others, specifically for the discussions on the best ways to perform research, to construct the research objectives and title, etc. They were not only contributing constructive ideas in my research work, but some of them have also read parts of my dissertation.

Finally, my heartiest gratitude goes to my family, to whom I give them my succesful as a gift, to my father late and to my soul (my mother), to my husband parents lates, to whom that they support me and suffer with me my respected husband (Msc. Sadaq Jebur), and deepest thanks to my intelligents sons (Jaafer and Osamah), and my nice and my lovely daughter (Tabarak), to who always believes me and prays for my successful and who are willing to extend a hand help, especially to my second father (The

Engineer Tareq Saleem), to my friendly brother (Dr. Ahmed Saleem), to my lovely and my faithful sister (The Engineer Aula Saleem), and to my honest friends. Lastly, for Allah, Alhamdulillah.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ABR	-	Adaptive Bit Rate
ADSL	-	Asymmetric Digital Subscriber Line
CCN	-	Content-Centric Networks
CDN	-	Content Delivery Network
CR	-	Content Router
CS	-	Content Store
FIB	-	Forwarding Information Base
HTTP	-	Hypertext Transfer Protocol
ICN	-	Information Centric Network
ID	-	IDentity
IP	-	Internet Protocol
IPTV	-	Internet Protocol Television
ISP	-	Internet Service Provider
LCD	-	Leave Copy Down
LCE	-	Leave Copy Everywhere
NAT	-	Network Address Translation
NDN	-	Named Data Networking
NS3	-	Network Simulation version 3
P2P	-	Peer-to-Peer
PC	-	Personal Computer
PIT	-	Pending Interest Table
Pprob	-	Path Probabilistic cache
Prob	-	Probabilistic cache
Rand	-	Random choice caching
RRT	-	Round Trip Time
TCP	-	Transmission Control Protocol

UCLA	-	University of California, Los Angeles
URL	-	Uniform Resource Locator
US	-	United State
VBR	-	Variable Bit Rate
VoD	-	Video on Demand

CHAPTER ONE
INTRODUCTION

This chapter provides an overview of the this research, including a background of the study, brief introduction of Named Data Networking (NDN) and its placement algorithms, and the online Video on Demand (VoD) architecture. The chapter also contains the research problem, research questions, and the research objectives. This will be followed by a brief explanation of the scope and significance of this research.

1.1 Background of the Study

The huge growth of the Internet has revolutionized the communication paradigms which include Named Data Networking (NDN), and an online video storage. The Internet Video on Demand (VoD) services use the existing and common Internet video architectures, such as HTTP and TCP [1, 2]. These are commonly used in YouTube, Vudu, and Netflix, due to their ability to stream video services to the third party commercial Content Delivery Networks or Content Distribution Networks (CDNs). The study of Psaras et al. [3] stressed that streaming of video over the Internet using HTTP has a lot of advantages: it is standardized across CDNs for portable video streaming service, it is universally accessible (CDNs had already made sure their service can reach through Network Address Translations (NATs) to end-hosts), and it is cheap (the service is simple, commoditized, and the CDNs competes on price). These benefits have made possible that the huge growth gives reasonable cost, high-quality movie and TV streaming, for the viewers' enjoyment [4].

The architecture of most commercial video streaming services is illustrated in Figure 1.1.

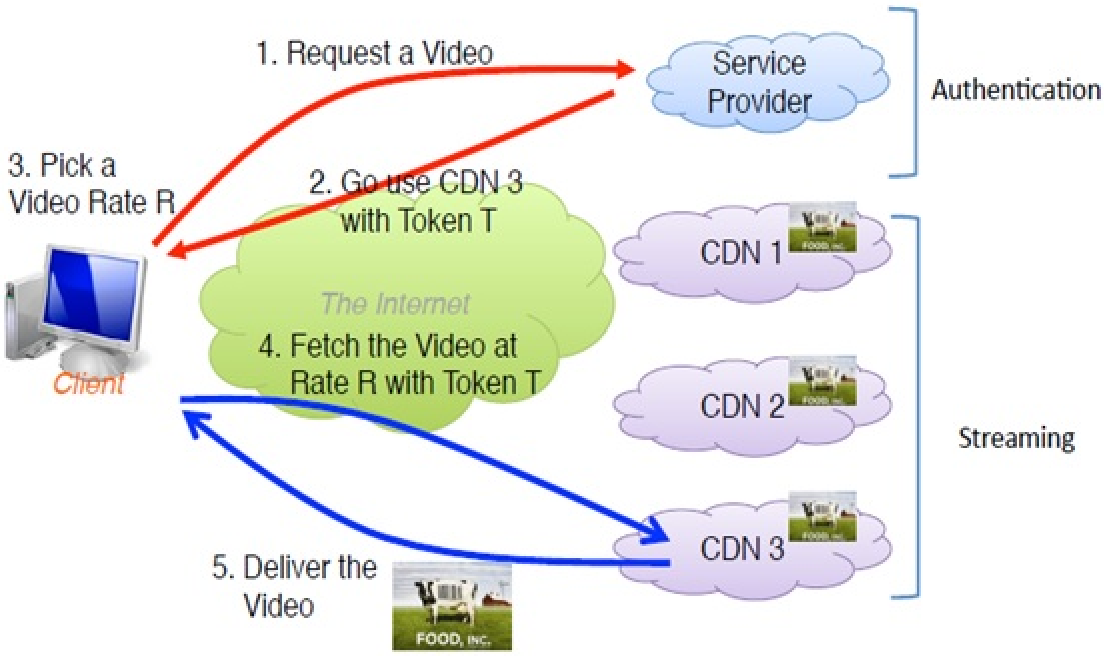


Figure 1.1: Architecture of Video Streaming Services over HTTP

As displayed above, video content is hosted at multiple CDN providers and streamed over HTTP to the clients. A video service generally supports several different platforms such as web browser plug-in, game console, and TV. A video session has two phases: authentication and streaming. When a client requests a video, the service provider authenticates the user account and directs the client to a CDN hosting the video. The video service provider informs the client about the available video streaming rates and issues a token for each rate. The client then picks a video rate and requests the video at the selected rate by presenting a token as a credential to the designated CDN [5, 6].

Internet video is commonly attributed with traffic related issues, unlike traditional video that needs to be completed prior to the commencement of playback. Previous studies have explained that some of the Internet videos are encoded at various rates ranging from 235 kb/second standard definition to 5Mb/seconds high definition

[7, 8].This leads to storing of the video as separate files on the server, thus, causing a lot of work for storing cache in NDN placement algorithms even though the NDN can make the strongest CDN [1]. As Internet video streaming has its wide storage location, sometimes failure is recorded due to the workload of the embedded data [9] .

The NDN design supposes hierarchically designed names, e.g., a video founded by the University of California, Los Angeles (UCLA) may have the name /ucla/videos/demo.mpg, where '/' delineates name components. The NDN architecture has two fundamental connection units: interest packet and data packet, both carrying hierarchically designed names [10]. When a consumer requests data, the interest packet is sent. A data packet can be utilized to fulfil an interest; therefore the name carried on the interest is a prefix corresponding to that of the data. An interest can also carry a selector field to determine priorities in case there are multiple data packets that can satisfy the interest [9, 10]. The architecture of a Content-Centric Networking (CCN) based Information-Centric Network (ICN) is illustrated in 1.2 below.

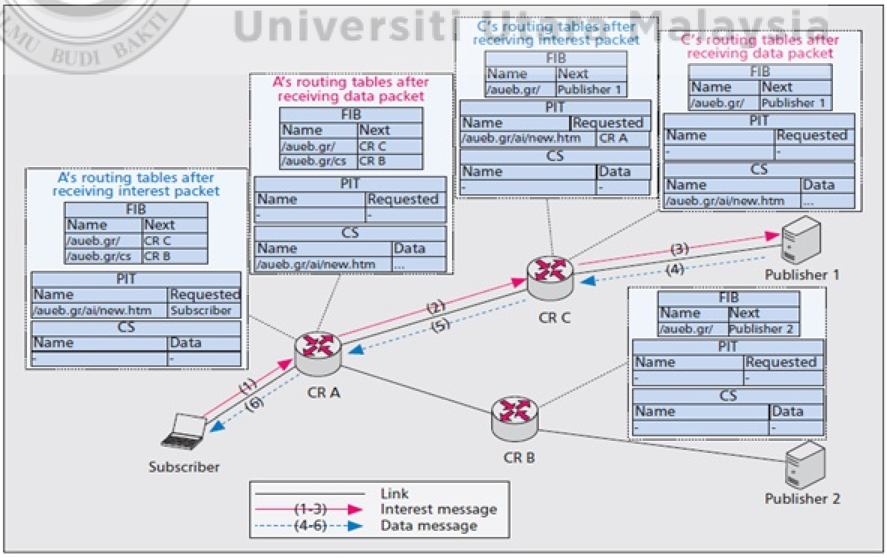


Figure 1.2: CCN Architecture

As shown in 1.2, CR refers to Content Router, FIB as Forwarding Information Base, PIT as Pending Interest Table, and CS refers to Content Store. However, Content-Centric Networking (CCN) is a receiver-driven communication protocol, which uses two distinct types of packets: interest and data packets. Each packet carries a name that identifies data which can be carried in one data packet. Each CCN content router (CR) contains three data structures: content store (CS) for temporary caching of receiving data packets, pending interest table (PIT) to contain names for interest packets and receive their matching packets, and forwarding information base (FIB) to forward interest packets [11, 12].

Figure1.2 also shows the workflow of the ICN architecture. An interest is sent by the subscriber of the name/aueb.gr/ai/new.htm (arrows 1–3). [When the interest file arrives, CR extracts the information name and searches for a matching file in its CS]. The interest packet is discarded if its matching prefix is found and then it is sent back through the incoming interface in a data message [11, 12].

In case there is no prefix matching the interest packet, a longer prefix match is performed by the router on its FIB in order to decide the forwarding destination of the packet. If an entry is found in the FIB, the router records the incoming interface of the interest packets in the PIT and pushes the packet to the CR as indicated by the FIB. When a prefix match is found at a publisher node or a CS, the Interest message is discarded and the information is returned in a data message. The CR stores the matched prefix of the received data message in its CS and then performs a longest prefix match in its PIT to locate an entry matching the data packet. If a PIT entry list multiple interfaces, it duplicates the data message and thus achieves multicast delivery. Finally, the CR forwards the data message to these interfaces while deleting the entry from the PIT (arrows 4–6). In case no matching entries in the PIT, the router discards the data packet as a duplicate [11, 12].

All communication in NDN is receiver-driven. A data packet can be received when a consumer sends data explicitly requesting an interest. When delivering an interest, routers increase the entry in the Pending Interest Table (PIT) [13], register the interface from which the interest is brought, and use a forwarding strategy to define where to send the interest. Therefore, a return data packet can easily trace the invert path back to the requester. This process basically builds a temporary multicast tree for each requested data item through which the data are efficiently received by all requesters.

Studies on the Internet traffic have shown that majority of tasks done online involve viewing, downloading, and uploading videos. Internet videos have been consistently growing for the last few years, with reports showing that around 51% of Internet traffic in 2011 was video. Market predictions suggest that video will comprise over 90% of the traffic on the Internet in the coming years. The increasing video workload is placing the onus on content providers for efficient distribution of the content [14, 15]. This implies that learning tools, tutorial, advertisement, and government activities are expected to be viewed or downloaded by an individual through the online videos. In other words, the availability of videos on the Internet prompts people to demand for the desired video, thus raising the issue of workload that cache offers at a specific time [14, 15].

In order to minimise ISP traffic, it is important to utilise effective cache placement algorithms which can also help in achieving better performance in NDN [16]. Among the most effective cache placement strategies, Leave Copy Everywhere (LCE) is the most widely used strategy which easily caches content data across routers [17].Leave Copy Down (LCD) is another placement algorithm used for multi-level caching to avoid replacement errors and unnecessary repetitions [16] . The content items are selected and cached randomly using the Rand placement algorithm. The issue of caching redundancy is one of the major challenges faced by caching strategies which can be reduced by a number of cache placement algorithms such as Prob and PProb [3], cross or hybrid caching [18], and centrality-based algorithm [19].

1.2 Research Motivation

The current utilization of different caches in computing systems is always considered to be the key method for addressing long memory access latency [1, 5]. This aspect has been widely discussed by different researchers in which the scaling of cache subsystems on-chip for supporting multiple layers is not necessarily sufficient for NDN networks. In addition, having several levels of customizing the resources' items may also introduce extra performance and design overheads [8]. This led previous studies to recommend the needs for efficient, scalable cache coherence protocols that can effectively operate in NDN cache by addressing the perspective for each cache placement algorithm when processing multiple copies of a cache line in multiple levels of the cache hierarchy [6, 8]. Therefore, the researcher here proposed to evaluate the performance of different cache placement algorithms to fully take advantage of the benefits of varied memory technologies.

1.3 Problem Statement

NDN architecture has been the rise in Internet video traffic, whereas this architecture makes communication easier between the receiver and request of online video [4]. This further enables the cache storage to easily deal with any form of online video requests simultaneously. Moreover, the forwarding strategy of NDN can detect and recover network faults independently. More particularly, it enables all the routers in the network to handle the network faults during online requests of video [4, 20]. The router usually responds promptly to user requests of online videos. However, the prompt response poses a serious challenge as different request often lead to delays in

VoD workload, which further affects the content cache of the router [4, 21].Giving a prompt reply to the user requests of online videos would serve as an activation for the performance of the network. Therefore, more responses to the different user requests of online videos would lead to delay in network performance.

The summarized problem statement of the this study is the issue of delay in VoD workload that is caused by the video traffic due to different user requests. The study will assess the NDN cache placement algorithms to evaluate their performance, and implement them.

1.4 Research Questions

This study aims at producing answers to the following research questions:

I. What are the reasons for the delay performance of NDN cache algorithms used in the VoD workload?

II. What are the key performance metrics to evaluate the performance of the NDN cache placement algorithms during VoD workload?

1.5 Research Objectives

The study's objectives are as follows:

I. To identify and analyze the delay performance in NDN cache algorithms used in VoD workload.

II. To evaluate the performance of the NDN cache placement algorithms using key performance metrics: delay, average hit ratio, total reduction in the network footprint, and reduction in load.

1.6 Research Scope

The focus of this study is to evaluate the performance of the NDN cache placement algorithms during VoD workload. The scope of this research is illustrated in 1.3 below.

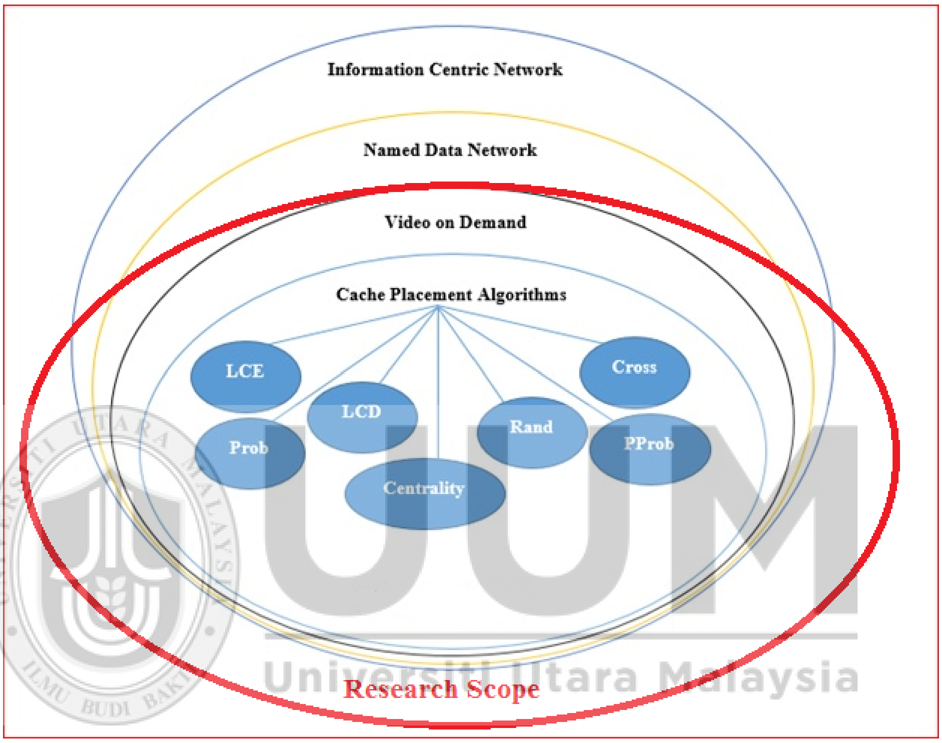


Figure 1.3: Research Scope

As shown in Figure 1.3, this study provides a broad outline of Information-Centric Networking (ICN) which is focused more specifically on Named Data Networking (NDN). The architecture of NDN includes an online video storage which stores the videos requested by consumers through the Internet Video on Demand (VoD) services. However, constant user requests often lead to delay in performance of the videos during VoD workload [4, 20].

In addition, the NDN has caching replacement and caching placement algorithms.

Whereas, caching replacement algorithms are out of this study's scope, the main scope of the study covers caching placement algorithms which is explained in details in chapter 2. Seven algorithms (LCD, LCE, Prob, Pprob,Rand, Cross, and Centrality) are presented, with the addition of Video on Demand service. This further demands the assessment of NDN cache algorithms to improve the slow response. This study will evaluate the performance of the seven cache placement algorithms using simulation and performance metrics during VoD workload.

1.7 Significance of the Research

The benefit of VoD is not limited to the aspect of science and technology, but also in other domains such as sociology, arts and government [21]. The countless benefits from using the Internet, especially the educational value, schools have now acquired the computer systems with telecommunication tools to have access to the Internet. In another study by Huang et al. [22], peer-assisted VoD services are considered as relevant as these videos ensure users to play the videos as the playback rate without any quality degradation while potentially reducing the publisher's bandwidth costs of video playback.

In addition, the study of Ukpebor & Emwanta [23] mentioned the academically, socially and morally driven benefits and implications: curriculum development of secondary school students, school policies on information retrieval, designers of search engines, and most importantly the teachers responsible for using the Internet as an educational tool in class. The impact of videos in education is far-reaching [24] as it develops interactivity with the content, which further helps learners in their cognitive development. VoD in real time enables learners to engage more with learning and more specifically with the flow of knowledge transfer and memory [24].

However, in case of large-scale content-centric networks, VoD services frequently face a number of challenges such as heavy video traffic, delay in VoD performance, constraints in content distribution, and so on. For instance, with the rapid increase in user requests for online videos, the storage capacity also gets increased which further makes replication of the entire storage data almost infeasible [25]. Moreover, constant responses to the user requests for online videos increase video traffic to an extent that it delays the performance of the VoD service and more particularly, affects content caching in the router [4, 21]. At the users' end, this further leads to decrease in service quality and high access delays due to long download times [26].

This study will focus on resolving the issues of online video traffic affecting both providers and users of the online video host. There would be a lot of revenue for the providers of service by Internet and indirectly the host of online video once the delay in responding to the request of VoD workload is reduced.

1.8 Outline of the Dissertation

The dissertation is divided into the following segments: **Chapter One** introduces the topic along with its background, statement of the problem, research questions and objectives. Also, the chapter contains the scope of the research, significance of the research and the outline of the dissertation. **Chapter Two** represents a review of previous related studies that focus on online Video on Demand and the NDN cache algorithm. Also, it explains the reasons of delay performance. **Chapter Three** emphasises on the methodological approach followed to achieve the research objectives. **Chapter Four** presents the results and discussions on the implementations of the seven cache placement algorithm using the NS3 simulator. Finally, **Chapter Five** concludes the work and suggests future works.

CHAPTER TWO
LITERATURE REVIEW

This chapter provides a detailed discussion of the extensive literature related to the Named Data Networking (NDN) cache algorithms. The chapter contains a detailed overview of NDN and its concepts, architecture of NDN, data structure of NDN and its operation, caching, the existing placement algorithms in the online video streaming, and the causes of delay performance in NDN cache algorithms used in VoD workload.

2.1 Named Data Networking and its Concepts

Researchers have argued that the Named-Data Networking (NDN) exemplifies the content-centric approach in networking [27, 28, 29, 30]. The location-independent content of the NDN is instantly addressable by a randomly long human-readable name, in spite of who performs it [31, 32]. The NDN has names and contents rather than interfaces or hosts, which it transforms into a first-class entity. In other words, the NDN provides each piece of content signed by the producer. This permits to decouple confidence of the content from the confidence of entities that must store and spread that content [29, 30].

Furthermore, the NDN possesses features that support the automatic content cache to enhance bandwidth usage and enable simultaneous successful usage of multiple network interfaces exemplifying the content-centric approach towards networking [27, 28, 29]. Previous studies have emphasized that the bandwidth use and effective simultaneous utilization of multiple network interfaces, improved by the NDN features, facilitate automatic caching of content [28, 29, 30]. This has added more values to the functionality and performance of the NDN in the networking domain [28, 29].

2.1.1 Reasons for the Need of Named Data Networking

One of the many efforts made to improve content-centric architectures are NDN [30]. It is associated with content-centric issues as it can combine revolting concepts of content-based routing [24, 30]. Moreover, the NDN has been emphasised to be built upon an open-source code-base called CCNx [24, 30]. NDN is one of the very few content-centric architectural suggestions with a rationally grown prototype ready in the networking domain [30].

Furthermore, the NDN is a contemporary research feature which is prone to continuous change [29, 30]. NDN denotes a reputable instance of content-centric networking design and at least some of its notions will affect the future of networking [13, 33]. Moreover, the NDN brings about some ideas, techniques and analysis that are used for an extensive range of designs, such as hosts, locations and content-addressable networks [29, 30].

In addition, NDN provides multicast of data which is one of the communication paradigms that have been researched in an online video storage. It potentially gives high priority to the data, including videos [10].

2.1.2 Named Data Networking Architecture

In NDN the receivers are driving the communication that is referred to as data consumers, through the swap of two kinds of packets: Data and Interest [28][27]. Both kinds of packets load a name that identifies a piece of data which can be sent in one data packet. The consumer's function is to put the name on a wanted piece of data into an Interest packet, and transmit it to the network [27, 28, 30, 33]. However, a unique name was used in routers in order to forward the Interest toward the producer of the data. During this process, caching strategies are relevant. The forwarding and caching in NDN is shown in Figure 2.1 below.

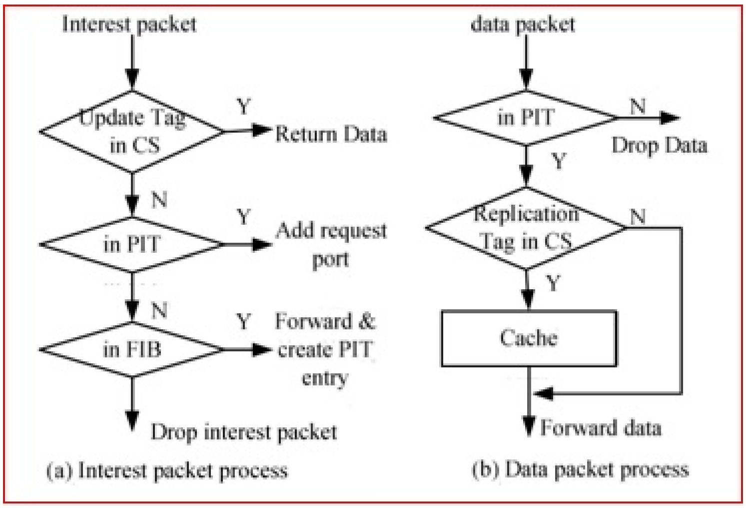


Figure 2.1: Packets in the NDN Architecture

As shown in Figure 2.1 above, after receiving the interest packet, a content router first checks for content availability in Content Store (CS). In case of a cache hit, CS sends the matched data back to the requester. Otherwise, the content router checks for content requests in the Pending Interest Table (PIT). PIT keeps track of the requested content items that are not yet served. In case of no entry found in PIT, a new entry is created in the Forwarding Information Base (FIB) which further forwards the interest packet to the interfaces. In case of a pending request found in PIT, the data packet of the target content is checked for its replication in CS. In case of replication, the data packet is forwarded to the requester through caching. Therefore, effective caching strategies are significant in NDN [31].

2.1.3 Limitation of the Named Data Networking

The lack of prime source or destination addresses in NDN facilitates privacy as long as NDN packets carry information about what is requested and not about the request maker [29, 34]. NDN designs poses four main important privacy issues to improve efficiency of networking [34].

I. The privacy of signature: Due to the nature of the digital signatures in publicly verifiable NDN content packets, the identity of a content signer may be prone to leak sensitive information.

II. The privacy of name: The content names of the NDN were motivated to be semantically compared to the content itself [34]. This is similar to HTTP headers where names significantly detect a lot of content information than the information of IP addresses. This would help an observer to simply determine when there are two requests that refer to the same content.

III. The privacy of cache: This ensures that the contemporary web proxies and network neighbours having the knowledge of the immediate content can be accessed by using the timing information in order to identify cache hits.

IV. The privacy of content: NDN gives permission to each entity with a name to receive symmetric content [29]. The encryption of the NDN is utilized to enforce access control and not for publicly available content. Therefore, the aim of the consumers is to receive the public content which cannot depend on encryption to hide the intended access.

2.2 Data structure in Named Data Networking

Previous studies have stressed that the basic data structures of the node model of NDN are as shown below [28, 30, 34, 35]:

1. Forwarding Information Base (FIB)

2. Pending Interest Table (PIT)

3. Content Store (CS)

The FIB at the table of forwarding is different from the FIB in IP routers and indexed by name prefixes in place of IP prefixes [30]. Every FIB entry provides multiple interfaces in place of a single better interface for each name prefix. Both PIT and CS are indexed by names, while a PIT entry record incoming and outgoing interfaces of an Interest guided by the data forwarding. Thus, a CS is a provisional cache of data packets which can speed up the satisfaction of Interests [35].

In addition, in the domain of the NDN forwarding process, the router delivers an Interest, checks the Interest name against the CS, and in case of a corresponding match, returns the data [30]. In the absence of this, the router checks the name of the Interest against the PIT to confirm if the Interest has previously been forwarded without signs of returning data. The study of Markopoulou et al. [34] emphasised that the router adds the Interest incoming interface to the entry of the PIT. If there is no PIT entry, the router adds a new PIT entry and searches for the Interest name in the FIB by using longer prefix match [35].

The occurrence of a matching FIB entry leads to the passing of Interest by a forwarding strategy module [28]. On the other hand, the router refuses the Interest and may transmit a NACK back to the incoming Interest interface [13]. Moreover, the router checks the data name of the received data packet against the PIT. Therefore, the existence of a PIT entry stores the data in the CS which is further forwarded to the incoming interfaces of the similar Interests that have been recorded in the PIT [13, 18].

In addition, the relevant forwarding strategy with the name of an area of Interest determines whether and how to forward the Interest [28]. This may take information such as the level of the routing protocol, the status of an interface, Round-Trip-Time (RTT), and the level of congestion in mind [36]. Moreover, they are assigned to each interface on each name prefix color code depending on the current work status [27, 28, 29]. The specialized color of the interface working is Green, while the Red color is set to non-working and Yellow to the uncertainty of the interface. Thus, the forwarding strategy always distinguishes Green interfaces from Yellow and Red interfaces that include forward Interests [27, 28].

2.3 Operations of the Named Data Networking

Evidences from the previous studies have shown that each network entity in the NDN provides content caching that are limited only by resource availability [9, 37]. This has the effect that the popularity of the content, which allows interest to be satisfied from the cached copies, distributed over a network and helped to maximize the use of resources [9, 38]. NDN is treated with authenticity and integrity of content by making digital signatures mandatory for all content packets. A signature linking content with the name supports the origin authentication regardless of its source [9, 29].

Moreover, NDN entities can publish new content producers, while referring those requesting to content consumers [39, 40]. Although the signature verification content is an election in the NDN, it is mandatory that the signature must be verified by any entity of NDN [39]. The studies of Ghodsi et al. [41] and Huang et al. [7] confirmed that the content packs carry additional metadata, such as identity (ID) for the publisher of the content and information about the location necessary for verification public key [41]. However, the NDN infrastructure does not impose a certain degree of confidence about trust management to individual application [7, 35].

Furthermore, restrictions and privacy in NDN is preserved through encryption by the content publisher [35]. The distribution of the encrypted content discovers that there is no mechanism for the application of a later encryption [7, 35]. Moreover, it can provide specific applications in a way to explicitly ask the content to be encrypted by publishers. However, the NDN currently refuses consumers to selectively hide content corresponding to their interests [28, 41].

The context of privacy about the absence of the source and destination addresses of the packets in the NDN is showing a clear advantage over IP [41]. This means that the discount that eavesdrops on the close link to the product content cannot be identified immediately on the consumer(s) who have expressed an interest in that content [30]. Besides, it features two standard NDN routing devices: caching content and the collapse of interests redundant, and the reduction of interest near tapping product of content since not all the interests of the same content are in its product [33].

In addition, studies have shown that the NDN supports any protection against an adversary that monitors local activity specific to the consumer since the names of most of the content are expected to be related linguistically to the same content, and can miss a lot of information about the content you wish to return [34]. NDN allows the use of "encrypted names", where the product encrypts the tail-end of the name [27, 33]. However, it does not provide a lot of privacy in activity networks, but the enemy can connect multiple interests to the same content [33, 34].

2.3.1 Routing roles in Named Data Networking

The forwarding plane design presented in assuming interfaces is ranked by routing preference [13]. In an extreme case, a strategic forwarding floods every Interest for all interfaces that can be obtained and implemented. Thus, data can be speedily retrieved always through the best paths, but, with significantly higher overhead. Besides, forwarding strategies exploring arbitrary interfaces or trying one after the other way round-robin can be implemented [28]. On the other hand, routers can find work since the paths are explored in all possible ways when given enough time [27].

2.3.2 Caching

NDN supports two approaches of caching: the caching of on-path and off-path with considering to the caches location [11, 42]. Both approaches of caching can be applied either separately or as a combination. The advantages of on-path caching against the off-path caching are stay under investigation [41, 42, 43]

1. **The on-path caching:** The NDN supports the on-path caching standardly, since each CR (Content Router) firstly consults its CS (Content Store) when it receives an INTEREST message and caches all information objects carried by data messages [11]. The caching decision is limited content propagated along the delivery path and to the nodes on the delivery path. So, on-path caching is integrated into the architecture itself. Caching is done at the network layer, thus, being independent of the application, but bounded by the online speed requirements of the receiving process, whereas the overhead of monitoring, collection of statistical information or notice of the cached content in a content notification service may not be agreeable or feasible. Lastly, on-path caching does not follow a specific topology [42].

2. **The Off-path caching:** It is provided by receiving an INTEREST to any data source that may be hosting the object of the requested information, e.g., the strategy layer can direct the INTEREST to the server of CDN rather than the originating publisher. This is not transparent to NDN whatever, as it requires populating the FIBs with pointers to such copies, which in turn requires the name prefixes of these copies to be noticed by the CDN server through the routing protocol used [11]. Off-path caching

leads to replicate content within a network, regardless of the forwarding path. Off-path caching is always centralized and requires a lot amount of information collected and noticed, in a content notification service. The problem of the ICN off-path caching is equivalent to the problem of replication defined in Content Delivery Networks (CDNs) and web proxies [11, 42, 43].

2.4 Concept of Video Streaming

Previous studies have revealed that video streaming calculated for 66% of total Internet traffic and calculated for over 40% of cellular traffic [35, 39]. This request has led cellular providers to run significant capacity to provide high quality of video streaming. Despite these efforts, performing dependable video streaming over cellular networks has proven to be complex [35]. In some cases, it has shown the stalled fraction videos raised with video quality with 10.5% of 240 pixel videos stalling whereas 45.7% of 720 pixel videos experiencing a stall [44].

Content providers these days use adaptive-bit-rate (ABR) streaming, whereas the aim is to correspond the delivery rate of chunks of the video to available end-to-end bandwidth [7, 35]. In other words, simple principle and practical implementations have to infer the obtainable bandwidth and adjust rates for chunks whereas balancing metrics like quality, interruptions and rate switch number [35, 38]. Besides that, the state-of-the-art algorithms do not comply at how they infer obtainable bandwidth and some use historical throughput, while others utilize buffer occupancy traffic shaping [45, 46].

Moreover, accurate inference of available bandwidth is non-trivial in order to differentiate the link capacities, congestion and some other factors [38]. This task is particularly challenging in cellular networks due to the inherent variability in signal strength, interference, noise, and user mobility causing the bandwidth to vary widely over time

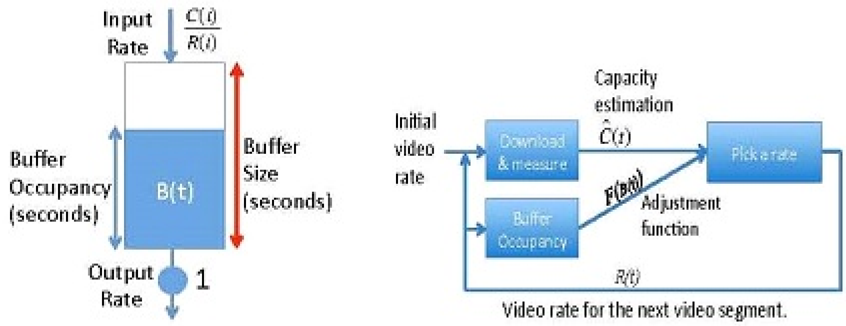
[46]. With proper outcome, there is a significant match between estimates used by existing algorithms and the natural obtainable network bandwidth that results in low quality of experience over cellular networks [45, 46]. Recent studies have opened a promising possibility, so as to accurately predict obtainable bandwidth at short and medium time scales [47, 48].

2.5 The Video Algorithms in Named Data Networking

The study of Huang et al. [7] shows formally that the algorithms below can avoid unwanted re-buffering events and yet perform a high average of video rate, and through a spread in the trade Netflix service. These common algorithms are the buffer-based algorithm and the baseline algorithm.

2.5.1 Buffer-based Algorithm

Although the motivation of finding the right modification of using buffer-based modifications in algorithms are completely attractive, the playback buffer is the exact state variable an Adaptive-Bit-Rate (ABR) algorithm is trying to do the controlling [46]. Researchers have emphasised that the simplest way to ensure that the merely request rate R_{min} is refused by the algorithm when the buffer methods are empty. Also, the algorithm gives permitting the buffer to increase as long as $C(t) > R_{min}$[46]. Such method simplicity would have avoided a rebuffing event [41]. However, as the buffer grows, it is safely increment $R(t)$ up to reach to the maximum video rate as the buffer methods are full as shown in Figure 2.2.



(a) *The relationship between system capacity, C(t), and video rate, R(t), in a video playback buffer.*

(b) *Adjust the estimation based on the buffer occupancy.*

Figure 2.2: The Dynamics of the Playback Buffer

In addition an ABR algorithm takes the rate of video as a function of the instance buffer occupancy, $B(t)$, so, it has buffer-based as shown earlier. The buffer-rate plane expresses the design space for this type of algorithms as shown in the Figure 2.3 below, where the rate-axis are the rate of video and the buffer-axis is the occupancy of buffer [41, 46]. So, the region between $[0, B_{max}]$ on the buffer-axis and $[R_{min}, R_{max}]$on the rate-axis identifies the practical region. On the other meaning, each curve$f(B)$ on the plane within the practical region knows a rate map, a function which has a video rate between R_{min} and R_{max} which produces the occupancy of instance buffer [41, 46].

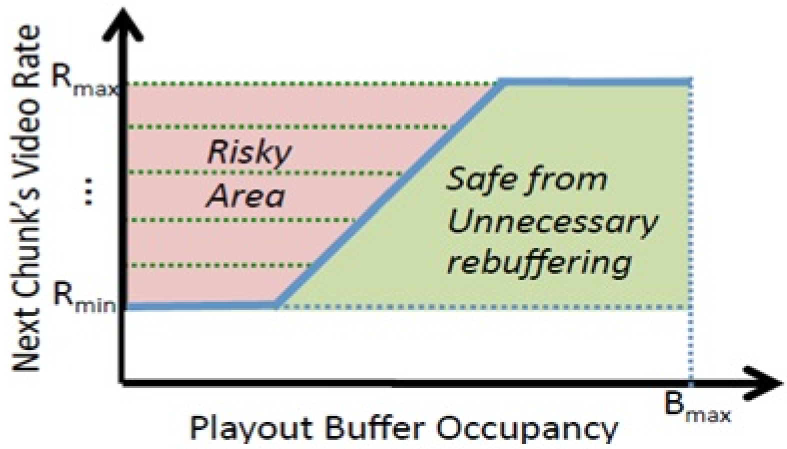


Figure 2.3: Video Rate As a Function of Buffer Occupancy

2.5.2 Baseline Algorithm

Testing of the buffer-based approach has to do with the construction of a baseline algorithm with some easy and naive rate map [35, 39]. Therefore, calling for the implementation of the algorithm in Netflix's browser-based player has a 240 second playback buffer and downloads the ABR algorithm at the beginning of the video session [39]. Although the player enjoys a larger buffer (240s) than embedded device players, it does not contain visibility or the network layer control [35]. This implies that a baseline algorithm requires setting the size of the reservoir to be 90s that would be big enough to handle Variable-Bite-Rate (VBR) [7]. Hence, maximization of the buffer distance between neighbouring rates is done, however, it leaves some room for the upper reservoir causes of setting the f(B) to be a linear function that arrives Rmax while the buffer is 90% full (216 seconds).

Moreover, the rate map does not completely identify the algorithm that shows the continuity of the rate map, while streamed video rates are separated, Rmin, R2, R3...Rm−1, Rmax. Thus, there is the need to adapt the rate following an easy rule at the current video rate as long as the rate supposed by the rate map does not cross the next higher (Rate+) or lower (Rate−) discrete video rate [45, 46]. Researchers have stressed that the buffer distance between the adjacent video rates produces a normal cushion to adapt this approach alongside with the rate map capable of developing the initial buffer-based algorithm [7].

2.6 Cache Placement Algorithms

A cache is a fast access memory which stores information that can be accessed promptly. Caching has been studied and applied in various domains in computer systems and more particularly in content-centric networks. A content placement algorithm is used for determining the sequence of routers from the original source of the content to the requester [4]. Effective cache placement algorithms are important for minimising ISP traffic and achieving better performance in NDN [16]. Some of the cache placement algorithms are described below. In order to evaluate the performance of the following algorithms, various researchers have used different performance metrics such as average hit ratio [49, 50], network footprint reduction [4, 51], and load reduction [4, 52].

2.6.1 LeaveCopy Everywhere (LCE)

Leave Copy Everywhere (LCE) is a widely used caching scheme in NDN. LCE supports that in NDN, each router caches all the content data that cross it. This caching system does not discriminate between its caching parameters. This further causes caching redundancy. More specifically, as discussed by Li et al. [17], when the population of content is increased, caching capacity is also enlarged. In the case of LCE, the caching performance remains stable, but there are other caching placement algorithms that outperform LCE. LCE is a good choice for flash-crowd events [4].

A copy of any content requested and delivered to the end user is duplicated at every router the information is traversing on its way to anyone. The advantage of this approach consists on its capability in producing caching redundancy, as various caches within the path are consuming cache resources to hold identical resources as discussed by Saino [53]. Meanwhile, the disadvantage of this approach is in its performance when processing resources in multiple-platforms.

2.6.2 Leave Copy Down (LCD)

In case of multi-level caching, Leave Copy Down (LCD) algorithm can be used for avoiding amplification of replacement errors and unnecessary repetitious caching of the same objects at multiple levels [16]. Moreover, the LCD can perform better under various workloads and interconnection topologies, especially while studying the caching behaviour of a multi-level cache [16].

It was mainly developed for hierarchical Web-caching techniques and builds advanced caching references in the event that a cache hit for just a particular content along the path. The main advantage of this approach is in its simplicity for replicating the content amount down to suit a particular cache size in certain path. This helps increase its visibility when the data is written towards the edge of a network. However, the disadvantage of this approach is that when data is written in multiple platform, a cache reference may not be sent to the source [54].

2.6.3 Random Choice Caching (Rand)

Random choice caching refers to the caching scheme in which a content item is randomly selected along the delivery path. It caches this randomly selected content item. This caching policy uses random selection of content items in order to evict the existing item and insert a new one. In terms of cache hits, this policy usually performs relatively poorly but it is used as baseline [4].

It catches a content resource in the event that items associated with certain packet are randomly selected along the delivery path. The advantage of this approach is in its feasibility in supporting multiple platforms in which the sequence of single resources is identified along the network path. However, the disadvantage of this approach in its number of nodes supported for multiple platforms [55].

2.6.4 Probabilistic Cache (Prob)

In the study of Psaras et al. [3], caching redundancy is a serious issue in decentralised and real-time content distribution in router caches. The study found that Prob cache, an in-network caching scheme, can reduce caching redundancy up to 20% in server hits, and up to 8% in the total number of hops that are required to hit cached contents. Thus, the Prob cache algorithm is effective in reducing network traffic redundancy to a considerable extent [4].

This approach is mostly used to enhance the distribution of nodes within the allocated packets for a certain network cache, that is certain, maximize the volume of distinct data began cached and combined for delivery to the source. The advantage of this approach is that subsequent of the data probability can be increased based on the referenced packets in the same path. However, the disadvantage of this approach is the possible delay when items of the packet is not sequenced based on the associated path [3, 4].

2.6.5 Pprob

In this caching algorithm, the probability of caching on a router is a function of its distance from the content origin and the shared storage capacity of the path [3].

The optimal algorithm has a cache miss when an item is requested. When the item under certain packet is requested s again, the PProb doesn't put the data for the requested item in the cache and doesn't evict any process till the reference is validated. Hence, the advantage of this approach is that it reduces the number of miss occurs when items are requested. On the other hand, it also reduce the time for replicating the items when it requested in future. However, the disadvantage of this approach is that it may require additional space for storing the sequence of items in different points on the cache [56].

2.6.6 Hybrid Caching (Cross)

Caching redundancy is one of the most serious issues among the contemporary caching algorithms. It usually occurs in the cached contents across different routers. Therefore, cache optimisation is required, which can be achieved by exploiting a cross-layer design based on content popularity and network topology in order to increase the cache hit rate and reduce traffic [18]. Studies revealed that a cross-layer design performs better in terms of increasing the overall cache hit rate and reducing network traffic [18].

It utilizes the assorted access latency involving cache sources, due for their physical destinations, to strengthen performance. The main advantage of this approach is in its cache banks can be of identical size for different routine technologies. The disadvantage of this approach is that it requires real-time access to items and large memory (more delay) when processing a higher number of items across different clients [57].

2.6.7 Centrality-based Algorithm

In the study of Chai et al. [19], a centrality-based algorithm was proposed to improve caching gain in order to perform better in terms of cache and server hit rates. The study exploited the concept of betweenness centrality in universal in-network caching. The results revealed that the centrality-based algorithm can achieve better caching gains in synthetic as well as real-time network topologies [4, 19].

In line with this approach, packets or data began processed are cached only once when the node utilize the greatest betweenness centrality (I. e. The node using the greatest volume of shortest walkways traversing it). The advantage of this approach is that multiple node has got the maximum benefit of betweenness centrality to which the content can be stored inside the node closer to the source. However, the disadvantage of this approach is when implementing it on dynamic systems, it can pose a problem in learning the node location based on its betweenness centrality in ego-network [19].

A comparison of these mentioned algorithms can be found in Table 2.1 from the table, it can be noted that most of these algorithms pose different characteristics which makes it difficult to judge its feasibility in certain network settings.

Aspects / Algorithm	Stability	Delay	Process multiple node	Feasibility in multi-platforms	Complexity
LCE	Provide stable performance when packet processed is in a single order.	Delay can be done when packets are requested from third-party network.	This algorithm is able to process multiple node within the same network.	Feasibility is low when it comes to process packet in multiple platforms.	Complex.
LCD	Is considered stable in most network scenarios.	Delay can be caused when cache size it increased.	The LCD is capable of processing multiple nodes across the ethernet.	The LCD is partially compatible to process packets through different platforms.	Moderately complex.
Rand	Is highly stable in peer-to-peer network activities.	It causes some delay when packet sent from one place to another due to its multi-layer for processing nodes.	Rand is not capable of processing multiple nodes when the reference is given to the original packet.	Rand may experience some difficulties when processing packets in multiple platforms.	Complex.
Prob	The Prob is mostly stable in distributed networks.	Delay may result when inconsistent data is began processed.	The Prob is capable of processing multiple nodes.	The Prob can deal with high packet loads in multiple platforms	Moderately complex.
PProb	Stable in most network settings.	Delay can be done when processing the multimedia sources with high cache sizes.	It allows multiple nodes to be stored.	It supports multiple platforms is the sequence of the reference for each node is constant with the packet.	Complex.
Cross	The Cross is not stable in distributed networks.	It may result in some delay when packets are not optimized.	Multiple nodes are supported by this algorithm.	It may faces some difficulties verifying the reference of nodes.	Complex.
Centrality	Centrality is not stable in most networks.	It can lead to a certain amount of delay when cache size increases.	Support multiple nodes.	Support multiple platforms.	Complex.

Table 2.1: A comparison study of the selected algorithms

2.7 The Causes of Delay Performance in NDN Cache Algorithms

The delay in the named-data network is caused by different events in which some were found to be associated with aspects related to the average time interval occurred usually when the generation of a request packet is affected by the traffic at the data source.

From this, some previous studies justified such cause to the queuing of service which relies mostly on the placement of packets from the source to the final destination. As such, certain delays could be caused by propagation and transmission to which the propositional logic for inductive delay is usually effected network overall performance. Such scenarios can lead to substantive changes on content naming and resolution that compress on routing, forwarding and security in-network caching and receiver-driven chunk-based network. This led some researcher [17] to investigate the feasibility of caching placement algorithms in improving the network performance. This can be achieved by conserving the required bandwidth, reducing data retrieval latency along with lessening optimal load invested by the server [58].

In addition, even with the use of placement algorithms, there may still a chance for having delay in a network due to the frequent storing of requested content in the network [59]. The named-data network along with content delivery network has been proven by the success of prevailing application which usually reserve good examples for data caching [60, 61]. Correspondingly, delay occurs while caching the contents at the selected server in NDN can lead to a sudden reduction in performance along with the access delay [62]. This was reasoned to that the subsequent requests does not require further transmitting to the content source, while then it is served by a closer NDN source along the server path.

The traffic size acts as another consequence of accessing delay in NDN where the content transfer across the ISP's domain boundary, which other external events related to congestion can be a reason why delay is taken place in caching with the concern of modifying the most popular content far from users [63].

On the other hand, a number of issues have already been addressed as the main antecedents effecting network handling access, data scalability, network management and mobility. Along with this, some considerations were given to the analyzed and integrated mechanism for managing popular content. One noticeable illustration is the need for delay/disruption tolerance, which usually acts as the main driver that may affect other network dominance [64]. Therefore, to ensure a fully utilized network, it is essential that delay related challenges are addressed [65, 66]. Those researchers explained the types of caching that promote reducing network delay while sending and receiving data which reasoned to the storage of data in a certain node when network faces sudden disruptions. These disruptions and delays could be caused by long distances which as a result the delay is caused. [58] The main events that could result in network delay based on the network failures that are particularly subject to the above types of events.

2.8 Summary

The previous work on the Name Data Networking and its concept in the Video on Demand domain was discussed in this chapter. It discusses the architecture of Name Data Networking, data structure, its operations, and caching placement algorithms. The contemporary algorithms in NDN, which is in line with the video streaming, while making requests of the consumers are reviewed. On the other hand, the slow response of the video streaming while making request is a critical issue that has not been enough addressed by the existing studies. Thus, there is the need to conduct further study on the possible approach to improve the response rate of the cache while requesting for the video online by using a placement algorithm.

CHAPTER THREE
RESEARCH METHODOLOGY

This chapter explains the detailed research methodology that uses to perform the objectives and includes the research framework, and research approach. Besides, the chapter also discusses evaluation procedures of the caching placement algorithms of NDN for the VoD services expressed simulation setup, the dataset sources, CloudTV content as VoD and the entries of operation in caching placement algorithms. Finally, this chapter includes the summary.

3.1 Research Framework

Churchill et al. [67] had earlier argued that the framework of the research could be regarded as the study plan or the design that leads to collect and analyze data. Meanwhile, researchers have stressed that the framework of the research guides and gives a structure for the compilation and checking the facts and reproduces rules about the precedence set to an array of the study procedure scope [68, 67]. Furthermore, De Vaus [69] argued that the prime purpose of a research framework is to ensure that the acquired evidence that enables by the researcher to answer the research question.

The framework of research is stressed by the previous studies as the number of stages of that represents the entire activities in the study, whereas, each stage has a number of steps. Therefore, this study has three stages of the certain activities as shown below which give rise to the expected results at the end of the activities:

I. Stage one:

It contains two steps, they are:

a. Explaining the study deals with the extensive analysis of the literatures leading to the formulation of research problems.

b. Doing study on the last experimental research activities and determine the causes of delay performance in NDN cache algorithms used in VoD workload.

II. Stage two:

It focuses on the experimental research activities with the aim to evaluate the performance of the NDN cache placement algorithms during VoD workload using key performance metrics: delay, average hit ratio, total reduction in the network footprint, and a reduction in server load by doing the following steps:

a. Implementing the seven NDN cache algorithms used in VoD workload to find the results of key performance metrics (delay, average hit ratio, total reduction in the network footprint, and a reduction in server load) on CloudTV content by using Network Simulation 3 (NS3).

b. Evaluating and analyzing the findings by using the experimental work and comparing the results of these algorithms based on cache size and writing of the report.

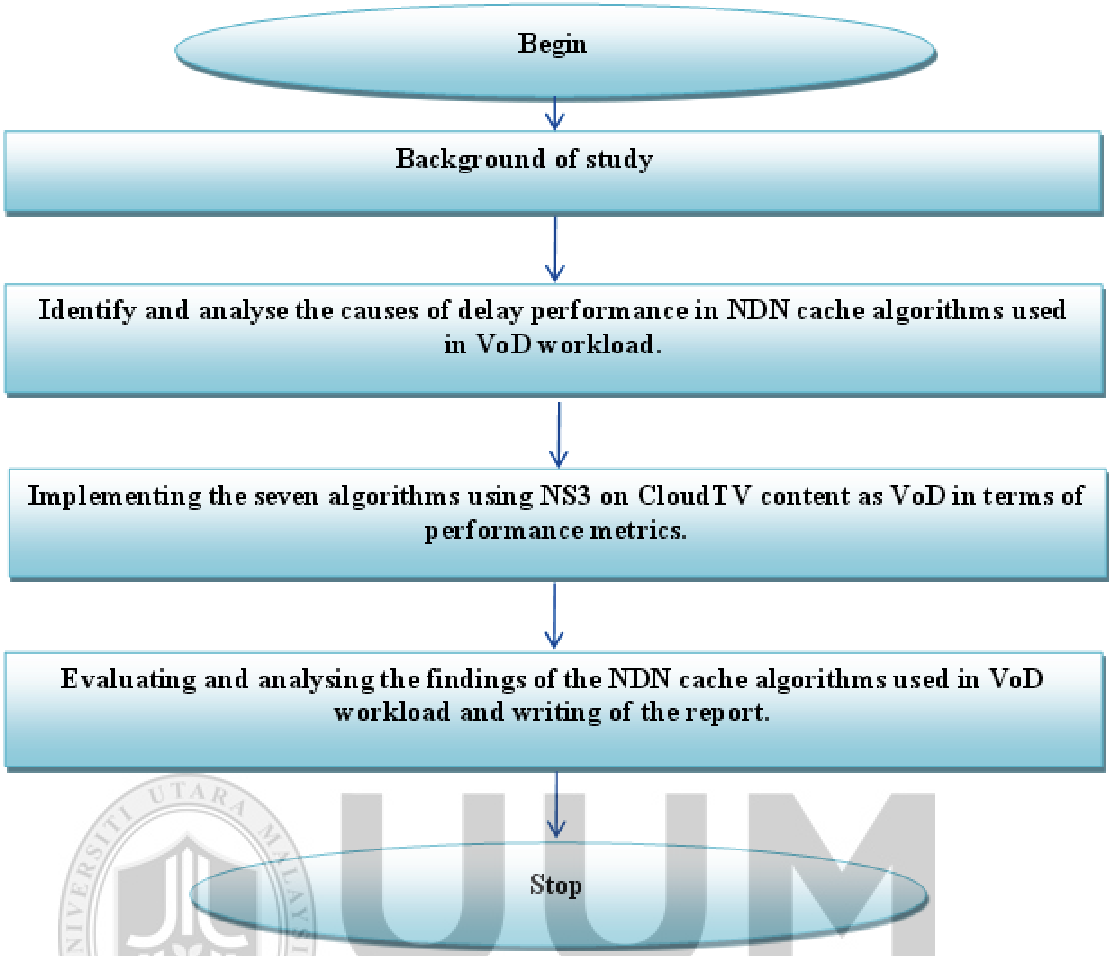


Figure 3.1: The Steps of research

The steps of this research are shown in Figure 3.1. These steps have been classified into two stages to achieve respectively the two objectives of this research.

In the first step, the study focuses on the issues about the NDN caching placement algorithms and VoD. This study provides sufficient background information and sets the context of this research. The second step is making study to determine the reasons of delay performance in NDN cache algorithms that deal with VoD workload that achieves the first objective.

The third step is implementing the seven NDN cache placement algorithms (LCD, LCE, Prob, Pprob, Cross, Centrality, and Rand) on CloudTV using NS3 simulator and finding the key performance metrics, which are the delay time, average hit ratio, total reduction in the network footprint, and a reduction in server load. Then, find the next step, that deals with analysis and evaluating the finding by using the parameters of video quality information about CloudTV content. So, this evaluation is done by comparing the results of these algorithms based on cache size that achieves the second objectives.

Whenever, The study of Sekaran [70] reiterated that the approaches are part of the framework which is in accordance with standards. While the study of Bryman and Bell [68] stressed that approaches are responsible to characterize data gathered. In addition, the study of Lancaster [71] stressed that the approach of the research can be collected into two; inductive and deductive. Therefore, the approach of the research describes the strategic presents of assumptions, ideas and techniques that elaborates the research study under observation [68]. In other words, Sekaran [70] emphasized that deductive reasoning is the research form approach applying a theory of testable or technique in the real world to evaluate their validity, while inductive reasoning is an approach of thinking whereby assuming the venues of an argument in order to maintain the conclusion, however, its assurance is not certain.

On the other hand, inductive reasoning centralizes on the observation of certain phenomena, which arrive at logical conclusions for proposing solutions [70]. Indeed, Sekaran and Roger [72] argued that deductive reasoning begins with a general technique or theory that can be done in a certain case under study. Therefore, this research will use the deductive approach since it intends to contribute the Name Data Network (NDN) algorithms of VoD services through the caching placement algorithms of online videos by implementing the seven cache placement algorithms in terms of performance metrics and comparing the performance of these algorithms based on cache size, this relates to the second objective.

All these steps are explained throughout this dissertation. The step one has been elaborated in Chapter one, but the second step will be discussed in Chapter two. However, the rest two steps will discuss in Chapter four, which include implementing the seven algorithms in terms of four performance metrics and evaluating the finding by comparing the results of these algorithms based on cache size. So, it will contain the details of the experimental setup, and the discussion on how the experiments. Finally, the conclusions and future works of this dissertation will be in Chapter five.

3.2 Research Implementation

This research implementation is performed by doing the experimental set-up to find the performance of the system. So, this study implements seven cache placement algorithms (LCD, LCE, Prob, Pprob, Cross, Centrality, and Rand) on CloudTV as a dataset and using NS3 simulator that leads to find the key performance metrics. Whereas, the video quality information is one of the parameters of the content of CloudTV [4].

3.2.1 Dataset Sources

The implementation of the caching placement algorithms in NDN for VoD service could be achieved in the context of this research by obtaining dataset which could be in the form of video streaming and a router level network topology. The online video streaming source would be the CloudTV as shown in Figure 3.2. The CloudTV is an application that is associated with online and real-time video streaming.

Figure 3.2: CloudTV Application

Moreover, CloudTV is another form of Internet Protocol Television (IPTV) application that has been accepted globally as a leading application for online content and green technology [5, 73]. CloudTV ensures real-time viewing, downloading of clips streaming, and commercial content to the consumers through a server box similar to P2P (Peer-to-Peer) platform delivery [20, 21]. Researchers have stressed that CloudTV application has larger than 52% of viewers across the Asia Pacific region [37].

3.2.2 CloudTV Content as VOD

This research would use the CloudTV content as shown in Figure 3.3, whereas the VoD is the source of the dataset form that requested from the IPTV server logs, while the server logs represent statistics of a single video request that transfers the log collection at the video session's end [21, 73]. The study of Sun et al. [4] highlights the operational categories of CloudTV in the common IPTV that have to be taken into consideration while sending request to the VoD services by using placement approaches in the NDN caching algorithm.

Figure 3.3: CloudTV Contents as VOD Services

3.2.3 NS3 Simulator

To implement simulation in the study, the used packet is an open source and implements NDN protocol stack for network simulation (NS3). Whereas, the address location of NS3 code is is (http://www.nsnam.org/ns-3-22/) [74, 75].

Moreover, the implementation of caching placement algorithm of NDN would be achieved by building scalable and extensive cache that could accept large streaming of video contents [21]. The NDN cache gives guarantee to complete simulation of a very big network with thousands of tens routers, NDN clients, and infinite number of content-view logs within a time reasonable period which would be lower than 5 hours. The NDN cache would provide an Application Programming Interfaces (APIs) to plug in placement algorithms by using NS3 simulation link.

After all, the researcher simulates these algorithms using NS3 version 3.22. Whereas, the Figure 3.4 shows the Internet2 topology utilized by the researcher in this study.

The researcher sets the Internet2 topology based on the link between direct-connected nodes in a bidirectional order which serve different purposes and their link utilizations generally have great differences. However, this Internet2 topology is used in this study because it is an exceptional community of the United State (US) and international leaders in research, academia, industry and government. Also, it supports the NDN environment and cache placement algorithms. In addition to it is an open source.

Figure 3.4: Internet2 Topology

3.3 Research Evaluation

The evaluation of the caching placement algorithms in the NDN that deals with VoD services seems achievable through fidelity, scalability and extensibility of the outcomes [40]. The fidelity of the algorithms which would be derived through the placement algorithm would give a venue for generations of several topologies and content request videos streaming [76, 77].

Furthermore, the evaluation procedures would be performed through the second objective, whereas the seven algorithms would be evaluated in term of performance metrics in the second objective and comparing them based on four cache size. In addition, the output of the simulation implementation to the algorithms will be measured, analysis and making comparison to evaluate them.The video quality information is one of the parameters of the content of CloudTV [4], as mentioned below: This evaluation is performed by doing the experimental set-up to find the performance metrics of the system

3.3.1 Experimental Setup

In this work, Video-on-Demand through CloudTV is used for evaluating the seven caching placement algorithms mentioned in the previous section. QoE (Quality of Experience) quantifies objectives by providing proper parameter measurements of the played video quality, or its subjective perception of the end user. The most important QoE parameters are buffering ratio, join time, and average bitrate. These parameters have an impact on the user engagement [54]. Buffering ratio is actually the fraction of the session time spent inside buffering and it was measured based on the following equation:

$$Buffering\,ratio = \frac{BufferingTime}{TotalViewTime} \tag{3.1}$$

The researcher measured the average bitrate based on the following equation [4, 54]:

$$Average\,bitrate = \frac{TotalBytesDownloaded}{TotalPlayTime} \tag{3.2}$$

In this study, the main focus was on analyzing the joins time for each of buffering ratio and average bitrate between the request and video start [4, 54], and is equal to:

$$JoinTime = StartTimeOfPlay - RequestTime \quad (3.3)$$

In this equation *RequestTime* variable refers to the time when the user requests the video from the CloudTV server. While *StartTimeOfPlay* variable refers to the time when the requested video plays after the user sends video request message to the CloudTV server. In addition, the entries of operations of CloudTV would be examined while evaluating the caching placement algorithms in the NDN for VoD services. Whereas, this experimental sends randomly CloudTV packets based on a trace file of an MPEG4 stream, which is high quality. This trace files could be downloaded from (http://www2.tkn.tu-berlin.de/research/trace/ltvt.html). A valid trace file is a file with 4 columns:

1- The first one represents the frame index.

2- The second one indicates the type of the frame: I, P or B.

3- The third one indicates the time on which the frame was generated by the encoder.

4- The fourth one indicates the frame size in byte.

In addition, the simulation link of NS3 would ensure a number of routers, clients, videos and many content requests, which provide scalability of the placement algorithm [77]. Hence, Table 3.1 that contains the simulation parameters, which is used to evaluate the seven algorithms.

Table 3.1: Simulation Parameter (CloudTV)

Description	Value
No. of node	50 nodes
Simulation time	400 *Sec*
Bandwidth	5 *Mbps*
Join Time	2 *Sec*
Cache size	1GB, 10GB, 100GB, 1TB
Packet size	1024*Bit*

3.3.2 Key Performance Metrics

Performance metrics are used for evaluating the performance of the cache placement algorithm for NDN during VoD workload. The following metrics are selected for measuring the performance of the cache placement algorithms: the delay time, the average hit ratio, total reduction in the network footprint, and reduction in load [4, 40, 78].

3.3.2.1 Delay Time

The delay is the total latency experienced by a packet to traverse the network from the source to the destination. The delay of end-to-end ($E2E$ Delay) path is calculated by doing the summation of the node delay at each node plus the link delay at each link in the path. Node delay includes the protocol processing time and the queuing delay at node i for the link (i, j). Link delay is the propagation delay on the link (i, j). In wireless link, the propagation delays are almost equal for each hop in the path [65, 79]. The each delay time is the different time between the request of the client time and response of the server time. As a result the delay time is calculated as follows [79]:

$$Average\,E2E\,Delay = \frac{\sum_{i=1}^{n}(R_i - S_i)}{n} \tag{3.4}$$

Where n refers to the number of requests of the client.

3.3.2.2 Average Hit Ratio

Average cache hit ratio across all routers can be defined by the following equation [4, 49]:

$$HitRate = \frac{\sum_{r=1}^{R} \frac{Hit_r}{Hit_r + Misses_r}}{R} \tag{3.5}$$

Where Hit_r refers to the number of requests that are served by the router r from its cache. Whereas $Misses_r$ refers to the number of requests forwarded upstream by the router due to cache miss.

3.3.2.3 Total Reduction in the Network Footprint

The total number of byte-hops saved through in-network caching can be defined by the following equation [4] :

$$TrafficReduction = \frac{\sum_{q=1}^{V} (HopsNoCache_q - HopsICN_q) * bytes_q}{\sum_{q=1}^{V} (HopsNoCache_q * bytes_q)} \tag{3.6}$$

Where $HopsNoCache_q$ refers to the total number of hops taken by the request in order to reach the nearest origin server. $HopsICN_q$ refers to the number of hops taken by the request to reach the closest location containing a copy of the video content, i.e., either an in-network cache or the origin server.

3.3.2.4 Reduction in Load

Reduction in load on the origin servers can be defined by the following equation [4] :

$$1 - \frac{NumCachedReq}{V} \tag{3.7}$$

Where $NumCachedReq$ refers to the number of requests that were served from some in-network cache.

3.4 Summary

This chapter explains the methodologies whereas this methodology is used for performing the research objectives. The methodology includes the detailed of research framework that used in the study. The stages of the research framework are discussed with their deliverables. The study assesses the existing NDN cache placement algorithms in terms of performance metrics during VoD workload. The performances of the placement algorithms are implemented using NS3, dataset sources, and CloudTV content as VoD in order to evaluate the seven cache placement algorithms. The used key performance metrics for simulation are a delay, average hit ratio, total reduction in the network footprint, and a reduction in server load.

CHAPTER FOUR
RESULTS AND DISCUSSIONS

This chapter provides result obtained from simulating algorithms LCD, LCE, Prob, Pprob, Cross, Centrality, and Rand on a CloudTV network using NS3. The chapter also elaborates on the performance metrics results related to these algorithms above.

4.1 Introduction

In the existing literature for Information-Centric Networking (ICN) architectures improvement metrics are well researched and studied, including the delay time, the average cache hit ratio, origin server load reduction, and the reduction in the overall network footprint [80]. This means that metrics could be optimized using various aspects of caching along with the content placement in order to make a decision which routers should cache the content on a request patch. In this work the researcher presented how different ICN caching placement algorithms impact on the main CloudTV performance metrics. The researcher evaluated seven cache placement algorithms, including four cache sizes of 1 GB, 10 GB, 100 GB, and 1 TB.

4.2 Simulation Scenario

In-network caching is very important in ICN architectures and there are a lot of ongoing studies on this subject that mostly falls into two main areas, content placement and content replacement [3]. In addition, current strategy associated with the settings of content placement indicates the cache of an object on the request path across the routers. For instance, a sequence of routers from the content origin of the requesting client must decide the cache of its nodes. The researcher firstly evaluated the cache placement algorithms on a CloudTV network. Analysis the algorithm results in terms of the delay in a network. In this study, the simulation scenario consists of seven particular content placement algorithms (LCD, LCE, Prob, Pprob, Cross, Centrality, and Rand).

4.3 Configuration of the Algorithms

Since the configuration of placement algorithms LCE, LCD and Prob shares certain aspects, the researcher considers the standard configuration [81] shown in Figure 4.1. It is well known that LCE caching placement algorithm is a typical mode of operation that is currently in use in most of the multi-level caches. In this study, the researcher configures the setting of LCE to provide a copy of the requested object when its cached in all intermediate caches (levels l-1,. . . , 1) on the path from the location of the hit down to the requesting client. This is configured when the hit happens at a level l cache or the origin server.

The LCD caching placement algorithm was set in this study to enable a new copy of the requested object to be cached only at the (l-1) -level cache. This was configured in order to allow cache to locate the requesting client by placing it below the location of the hit. A semi structures were used in LCD similar to LCE because it involves multiple requests for the aim of capturing object to a leaf cache. It was also set to locate each request by creating a new copy of the object one client's hop.

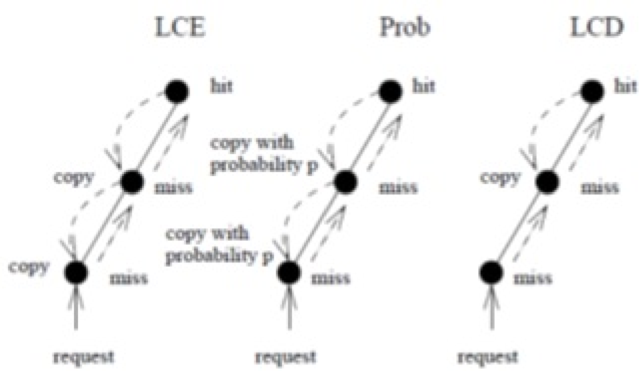


Figure 4.1: Operation of the LCD, LCE, and Prob Cache Placement Algorithms

As for the Rand caching placement algorithm, the researcher, configures Random's caching properties to select one of the nodes on the trajectory randomly. Moreover, it was set to place a copy of content file on a randomly designated node, whereas its reference packet is processed through the network to build up the potential field for node caching. In the simulation, the researcher considered sending random requests within a predefined range to direct the distribution of the content file, and the query to the associated node. This was necessary to allow forwarding queries within certain node randomly until it locates the potential field.

As for the Prob caching placement algorithm, the researcher configures the placement of caches by regulating the path from the location of the hit along with the referenced query associated with client's intermediate cache is appropriate. The researcher considers placing a reference object for it to be used in identifying the origin of client's request. On the other hand, local copy with probability p is kept from the intermediate cache, by doing so; replacement algorithm is invoked as a result. This can be reasoned to that copy of the present work was not placed in the intermediate cache with probability 1-p [81].

The Pprob caching algorithm is a probabilistic scheme for distributed content caching along a patch of caches. In this study, the researcher configures Pprob in order to provide an estimation of the caching capability based on the node path location and cache contents probabilistically suggested [81] as shown in Figure 4.2. As a result of this it leaves caching space for other flows sharing the same path, and fairly multiplex contents of various flows among caches of a shared path.

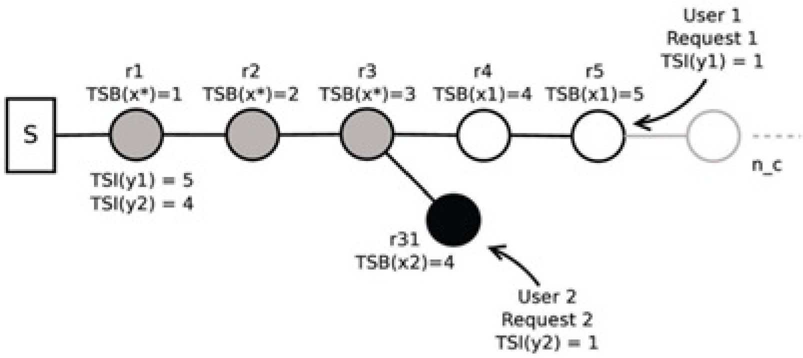


Figure 4.2: Scenario of Pprob Caching Algorithm from

Centrality-based caching algorithm was used in this study to enhance the caching gain and reduce the ambiguity in the performance of node placement. The researcher considered setting provided by Chai, He, Psaras, and Pavlou [19] to set the number of times of predefined node on the content delivery path by considering the association that links pairs of nodes in the network as shown in Figure 4.3. By doing so, the researcher in this study assumed that the content node may drop in the event of a high number of content delivery paths, which reasoned to the cache misses within a network. In addition, the researcher ensured that cache replacement rate is condensed by only caching nodes with higher priority along with caching content.

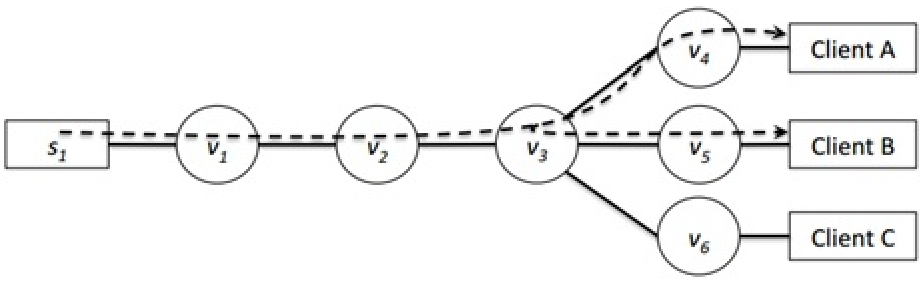


Figure 4.3: Node Placement in Centrality-based Caching Algorithm

Cross caching placement algorithm was set in this study in a cross-layer form for the aim of caching contents in a few selected destinations by considering the relationships exists between content popularity and the network topology. The researcher sets the algorithm to exploit available information at both network and application layers.

4.4 The Results of Performance Metrics

A number of trends related to the performance of networking applications can help to enhance the Quality of Service (QoS) supported across multiple environmental conditions [78, 82, 83]. In this study, the researcher considered certain network measurements for composing network metrics of the algorithms.

4.4.1 Delay Time

The each delay time is the different time between the request of the client time and response of the server time. As a result the delay time is calculated as follows [79]:

$$Average\,E2E\,Delay = \frac{\sum_{i=1}^{n}(R_i - S_i)}{n} \tag{4.4}$$

Where n refers to the number of requests of the client.

The delay performance of the selected algorithms was examined and compared in different cache sizes such as 1GB, 10GB, 100GB, and 1TB. This was necessary to enrich the current results obtained from the previous performance metrics. In addition, the comparison was following the standard setting through which the sent packet was treated as the main source for data transfer.

Figure 4.4 shows the delay performance for all algorithms when the cache size is 1GB. It can be noticed that the algorithms of CENTER, CROSS, and RAND result in high delay performance of 4.824798 Sec. However, LCE was found to be the best delay which it results in 4.781457 Sec delay performance followed by Prob (4.784552 Sec) and Pprob (4.791872 Sec). This can be reasoned to that LCE consists of a queue editor in which the processed packets are sent in a constant order [4, 17]. This led the researcher to conclude that LCE, then Prob to be the most sufficient algorithms when it comes to delay performance in cache size 1GB.

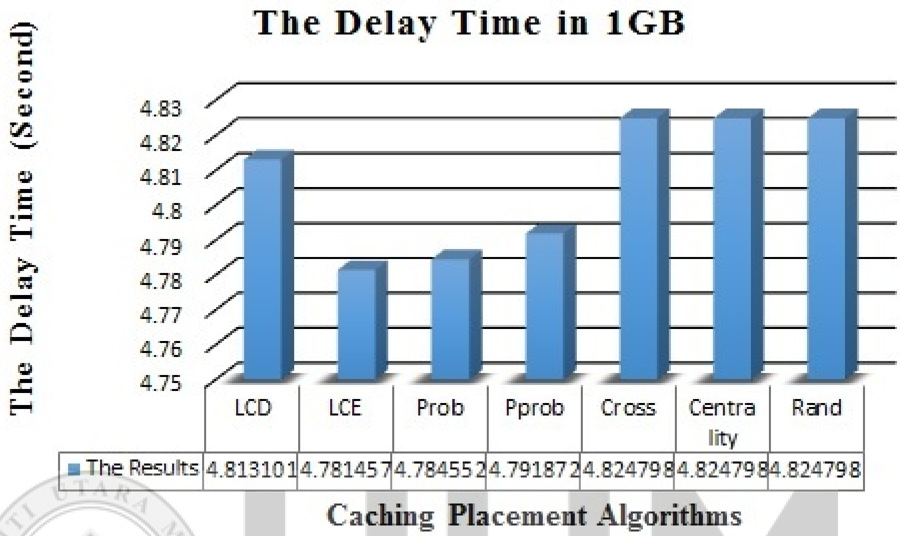


Figure 4.4: Delay for all algorithms with 1GB

On the other hand, Figure 4.5 shows the delay performance for all algorithms when the cache size is 10GB. From the result, one can notice that the same three algorithms of CENTER, CROSS, and RAND have again resulted in high delay performance of 4.717 Sec. However, Pprob was found to be the lowest in delay performance with 4.669187 Sec followed by LCE (4.677418 Sec) and then Prob (4.684028 Sec). This can be reasoned to that Pprob in higher cache sizes apply effective queue management, whereas the packet is pre-processed based on the number of nodes and priority [10, 81]. This led the researcher to conclude that Pprob and LCE are still considered the most sufficient algorithms when it comes to delay performance in cache size 10GB.

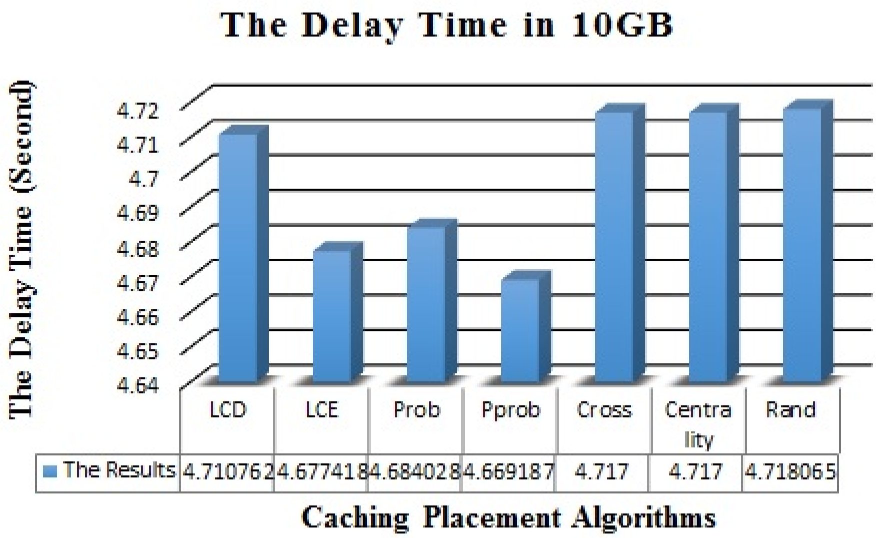


Figure 4.5: Delay for all algorithms with 10GB

Meanwhile, Figure 4.6 shows the delay performance for all algorithms when the cache size is 100GB. The performance result revealed that algorithms of CROSS (delay= 4.70809 Sec) and RAND (delay= 4.683788 Sec) have resulted in high delay performance. However, Prob was found to be the lowest in delay performance with 4.5638 Sec followed by Pprob (4.599327 Sec). This can be reasoned to that Prob is useful when it comes to process higher cache sizes because it distribute the packet into levels that can be recalled and processed faster by the network [3, 81]. This led the researcher to conclude that Prob and Pprob are still considered the most sufficient algorithms when it comes to delay performance in cache size 100GB.

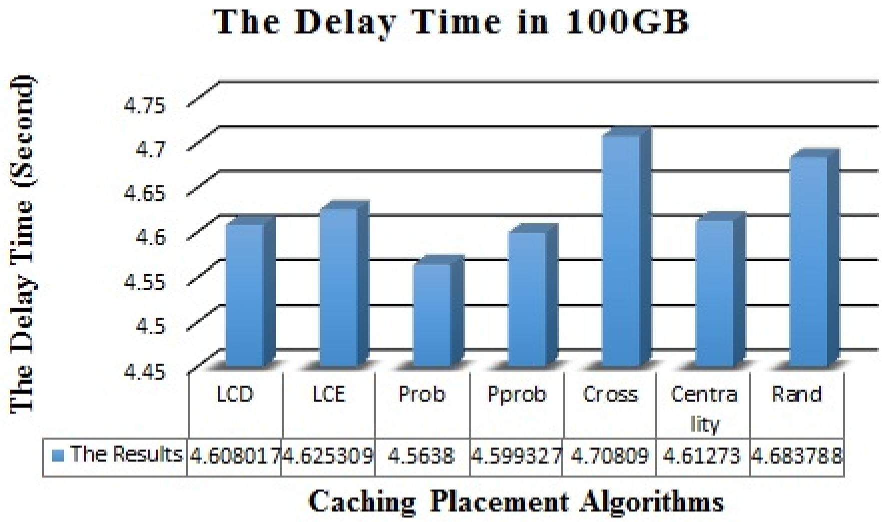


Figure 4.6: Delay for all algorithms with 100GB

Meanwhile, Figure 4.7 shows the delay performance for all algorithms when the cache size is 1TB. The performance result revealed that same algorithm of CROSS (delay= 4.637841 Sec) and RAND (delay= 4.612946) have resulted in high delay performance. However, Prob again was found to be the lowest in delay performance with 4.51102 Sec followed by Pprob (4.51352 Sec). This can be similar to the reason mentioned above where Prob in huge cache size distribute the packet into levels that can be recalled and processed faster by the network [3, 81]. This led the researcher to conclude that Prob and Pprob are still considered the most sufficient algorithms when it comes to delay performance in cache size 1TB.

Table 4.1: The Delay Time in Four Cache Sizes

Cache Size	LCD	LCE	Prob	Pprob	Cross	Centrality	Rand
1GB	4.8131013 sec	4.781457 sec	4.784552 sec	4.7918717 sec	4.8247982 sec	4.8247982 sec	4.82479819 sec
10GB	4.7107619 sec	4.677418 sec	4.684028 sec	4.669187 sec	4.7170003 sec	4.7170003 sec	4.71806454 sec
100GB	4.6080171 sec	4.625309 sec	4.5638 sec	4.599327 sec	4.70809 sec	4.61273 sec	4.683788 sec
1TB	4.53009 sec	4.551379 sec	4.51102 sec	4.51352 sec	4.637841 sec	4.571125 sec	4.6129463 sec

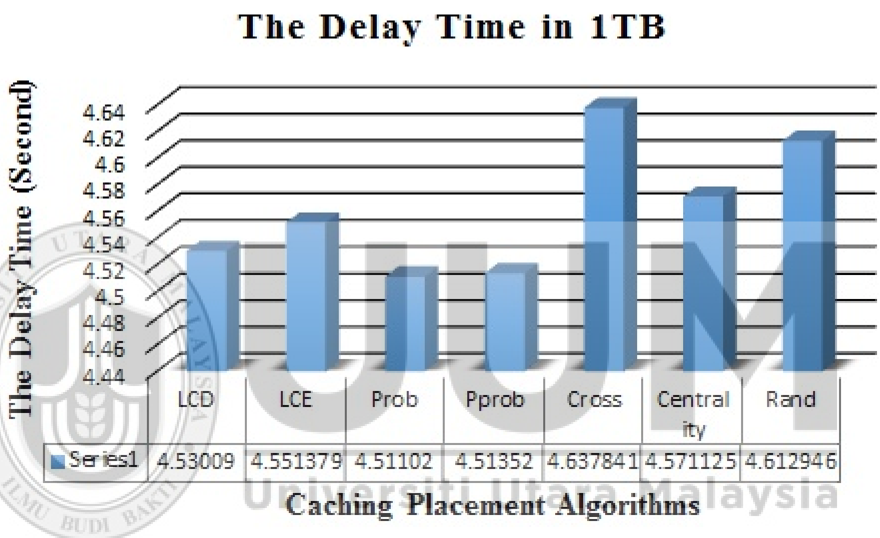


Figure 4.7: Delay for all algorithms with 1TB

Furthermore, Figure 4.1 explains the delay for all algorithms in four cache sizes as in Table 4.1, therefore, it can conclude that CROSS and RAND algorithms have approximately the worst delay values in the 1GB cache sizes. When the LCD gives the best delay performance with 1GB. While Pprob algorithm obtains the best algorithm in 10GB. Meanwhile, Pprob has the least delay time in 100GB and 1TB. However, the reasons for these results of delay are explained previously above.

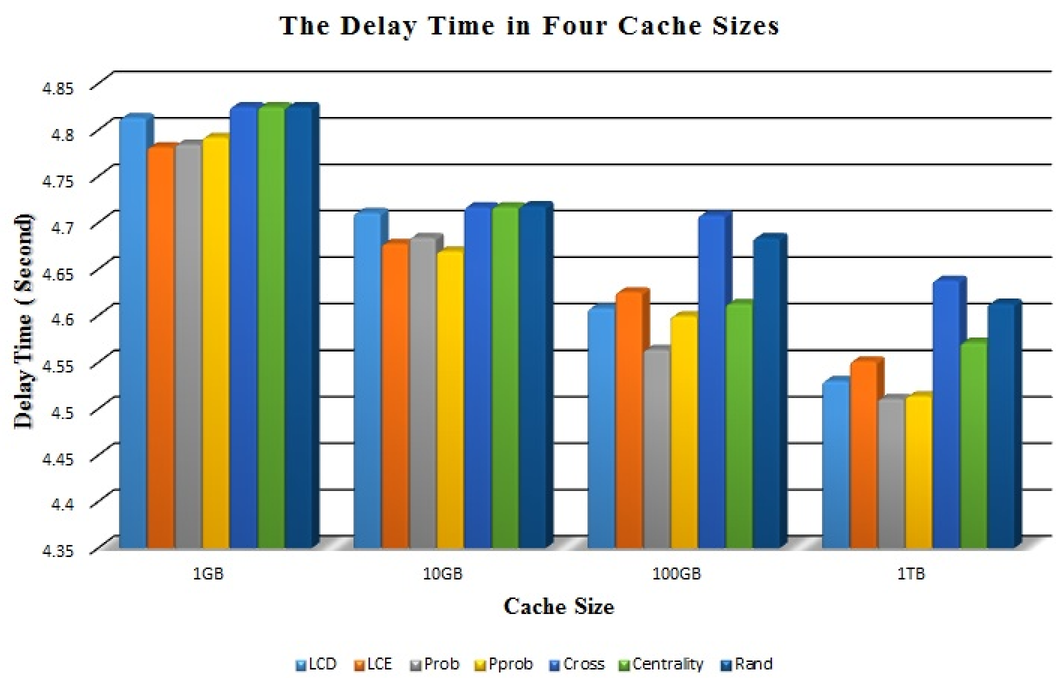


Figure 4.8: Delay for All Algorithm in Four Cache Sizes

4.4.2 Average Cache Hit Ratio

Average cache hit ratio across all routers was measured in this study based on the following equation:

$$HitRate = \frac{\sum_{r=1}^{R} \frac{Hit_r}{Hit_r + Misses_r}}{R} \tag{4.5}$$

In the above equation $Hitr$ presents the number of requests that router r was capable to serve from its cache, and $Misses_r$ presents the number of requests that the router forwarded upstream because of the cache miss.

In the next diagrams we will present the average cache hit ratio of all seven caching placement algorithms for cache sizes of 1GB, 10GB, 100GB, and 1TB [4]. In this way we will compare the average cache hit ratio results between the seven cache placement algorithms for different cache sizes.

Figure 4.9 presents the results of the average cache hit ratio of all algorithms when the cache size is 1GB. However, Pprob and Centrality had also a similar average cache hit ratio of 0.004. While, the others algorithms range between 0.007 and 0.006. Finally, the result showed that LCD has the highest ratio of 0.01 as compared to other algorithms.

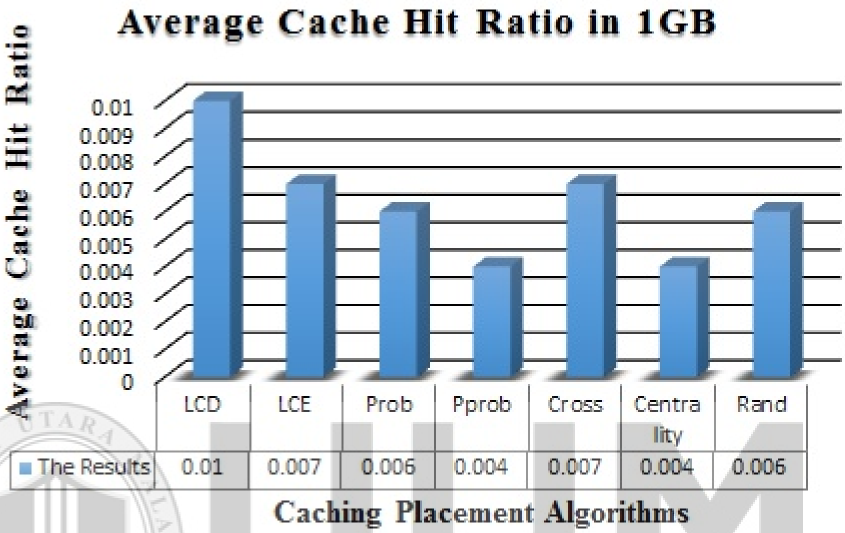


Figure 4.9: Average Cache Hit Ratio for All Algorithms in 1GB

In Figure 4.10, the cache size is increased to 10GB. We can already notice here that the average cache hit ratio of the LCD algorithm gives the best performances. It has the highest value of the average cache hit ratio that is 0.023 followed by Rand (0.016). However, other algorithms such as LCE, Prob, Pprob, and Cross achieved interchangeable ratio, whereas Centrality resulted in lowest ratio in 0.008. Centrality algorithm again gives the lowest value for the average cache hit ratio.

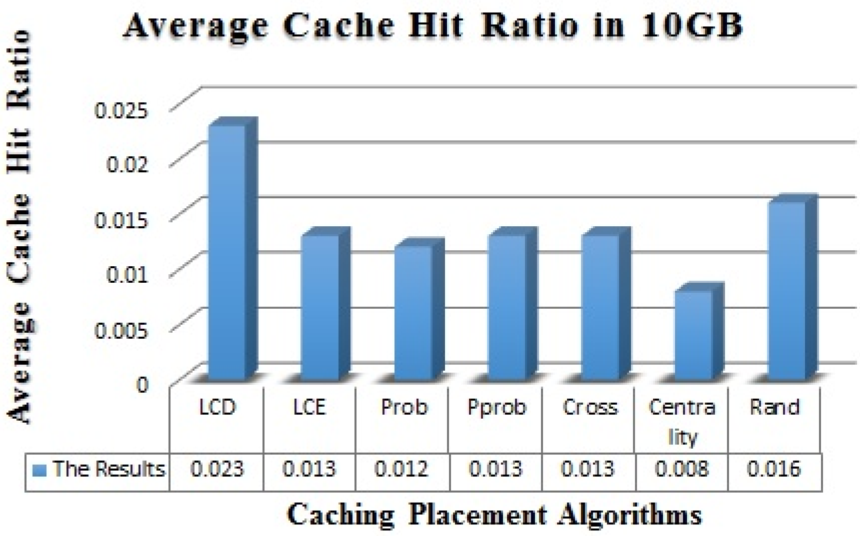


Figure 4.10: Average Cache Hit Ratio for All Algorithms in 10GB

In Figure 4.11, the cache size is increased to 100GB. It can be concluded that an average cache hit ratio of the Rand algorithm give the highest performances 0.048 followed by LCD 0.047, and Pprob 0.046 respectively. However, other algorithms such as Centrality resulted in a very low ratio of 0.023 as compared to Cross, LCE, and Prob.

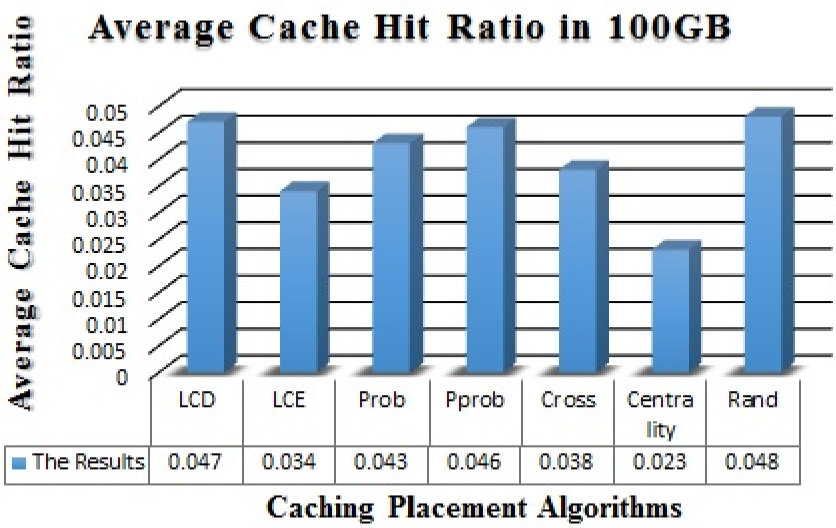


Figure 4.11: Average Cache Hit Ratio for All Algorithms in 100GB

Figure 4.7 presents the average cache hit ratio of all seven caching placement algorithms when the cache size is 1 TB. Prob and Rand have the highest value for the average cache hit ratio of 0.091 as compared to the other six algorithms. Just LCE and Pprop algorithms have an average cache hit ratio that is little less than the Prob algorithm. Centrality algorithm again gives the lowest results for average cache hit ratio of 0.043.

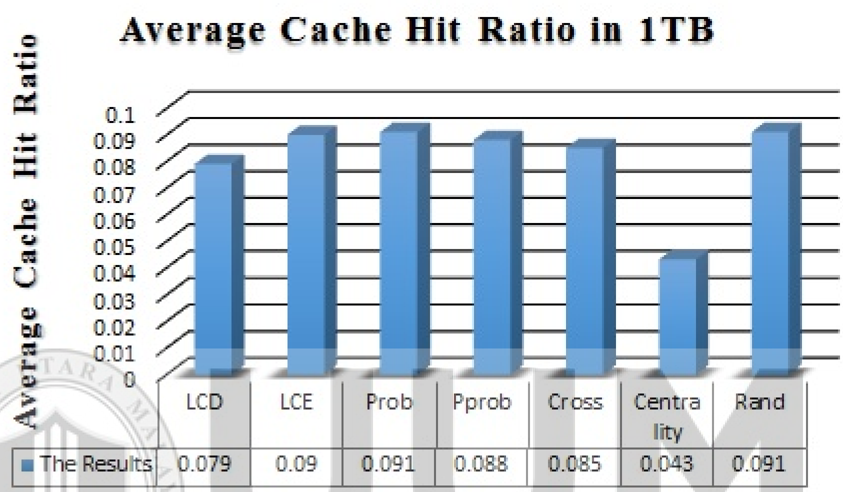


Figure 4.12: Average Cache Hit Ratio for All Algorithms in 1TB

Generally, in all diagrams from Figure 4.9 to Figure 4.12, the average cache hit ratio increases in all cases for all seven caching placement algorithms as the cache size increases. We can also conclude that Centrality algorithm gives almost the worst result of cache hit ratio as shown in Figure 4.13 below that elaborates the results of the average cache hit ratio in four cache sizes as shown in Table 4.2. Whereas, LCD algorithm gives the best results of the average cache hit ratio using both cache sizes (1GB, and 10GB). When, Rand algorithm has the best result by using 100GB and 1TB. In addition, Prob algorithm gives the best value with 1TB only.

Table 4.2: The Average Cache Hit Ratio in Four Cache Size for placement Algorithms

Cache Size	LCD	LCE	Prob	Pprob	Cross	Centrality	Rand
1GB	0.01	0.007	0.006	0.004	0.007	0.004	0.006
10GB	0.023	0.013	0.012	0.013	0.013	0.008	0.016
100GB	0.047	0.034	0.043	0.046	0.038	0.023	0.048
1TB	0.079	0.09	0.091	0.088	0.085	0.043	0.091
Average	0.03975	0.036	0.038	0.03775	0.03575	0.0195	0.04025

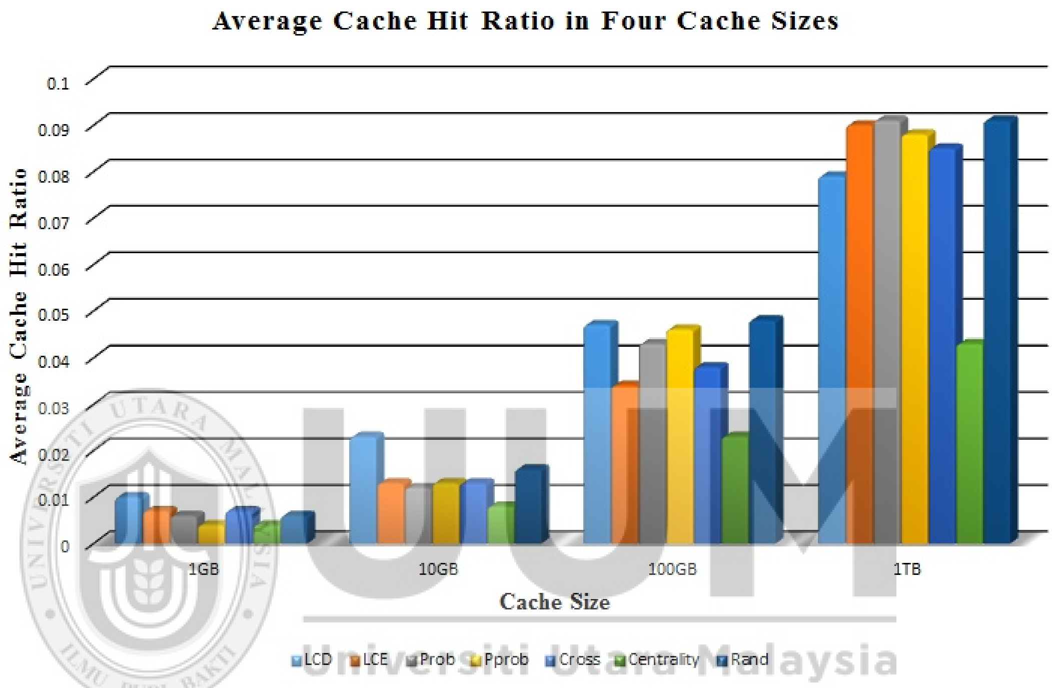


Figure 4.13: Average Cache Hit Ratio of the Cache Placement Algorithms

However, we can see that LCD is better than LCE, in spite of, LCE caches a copy of content on every node on the path, while LCD stores it on each node on the level below where the hit occurs. Furthermore, both LCD and LCE used a semi structures because they require multiple requests in order to caption object to a leaf cache. They also set to locate each request by creating a new copy of the object one client's hop. Besides that, N. Laoutaris et al. [16] focused that the LCD is better than LCE whereas, the LCD is able to support for exclusive caching. In other words, it allows each cache on a chain of caches to hold a potentially various set of objects, so the LCD avoids the repetitious replication of the same few objects. The previous studies [82, 83] concluded that the cache strategies approximately performed an optimal cache content distribution by decreasing the redundancy of cache content (however LCD has decreased redundancy than LCE), that leads to increase the variety of content strategy, therefore this lead to give effectively increased hit ratio. Also, the studies [4, 82, 83] found that the LCD had an average cache hit ratio larger than LCE.

4.4.3 Total Reduction in the Network Footprint

Total reduction in the network footprint is in fact the total number of byte-hops that are saved via in-network caching [4]. It was measured based on the following equation:

$$TrafficReduction = \frac{\sum_{q=1}^{V}(HopsNoCache_q - HopsICN_q) * bytes_q}{\sum_{q=1}^{V}(HopsNoCache_q * bytes_q)} \quad (4.5)$$

In the above equation $HopsNoCache_q$ refers to the number of hops the request would take to get to the nearest origin server. $HopsICN_q$ presents the number of hops that the request would take to reach the closest video content provider either an ICN cache or the origin server.

In the following diagrams we will present the values of the reduction in the network footprint for all seven caching placement algorithms for different cache sizes in order to evaluate the algorithms by comparing the results of them.

Figure 4.14 presents results for the reduction in the network footprint for all algorithms when the cache size is 1GB. We can notice here that the LCD algorithm has the highest value of a reduction in the network footprint 3.6. While other algorithms such as Cross and LCE has similar reduction value of 2.5. The lowest reduction in the network footprint gives Centrality and Pprob algorithms.

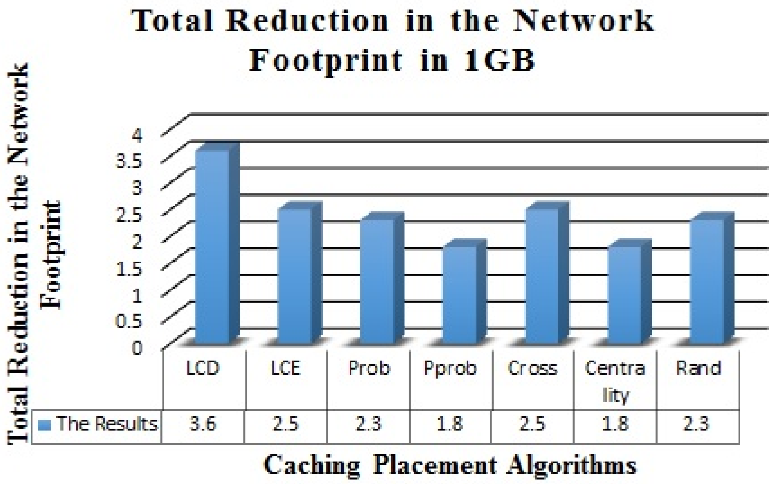


Figure 4.14: The Total Reduction in the Network Footprint for Caching Placement Algorithms in 1GB.

Figure 4.15 shows the results for the reduction in the network footprint for all algorithms when the cache size is 10GB. LCD algorithm again gives the highest value of a reduction in the network footprint together with the Rand algorithm. Centrality algorithm again gives the lowest value of the reduction in the network footprint. Comparing to the footprint in 1GB, all seven algorithms have higher values for the reduction in the network footprint, because the cache size is increased from 1GB to 10GB.

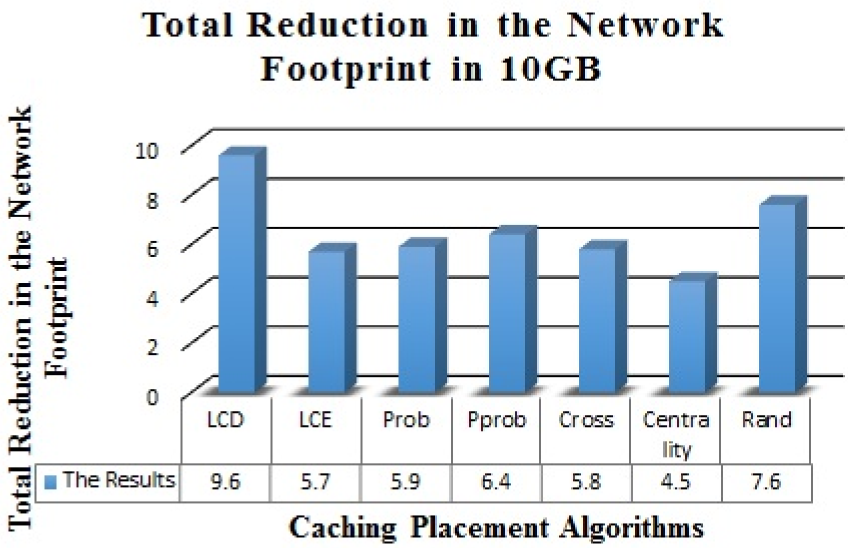


Figure 4.15: The Total Reduction in the Network Footprint for Caching Placement Algorithms in 10GB

Figure 4.16 depicts the total reduction in the network footprint for all algorithms when the cache size is 100GB. LCD algorithm again gives the highest reduction in the network footprint, followed by Rand, Prob, Cross, Pprob, and LCE respectively. However, Centrality algorithm was the lowest in the footprint reduction.

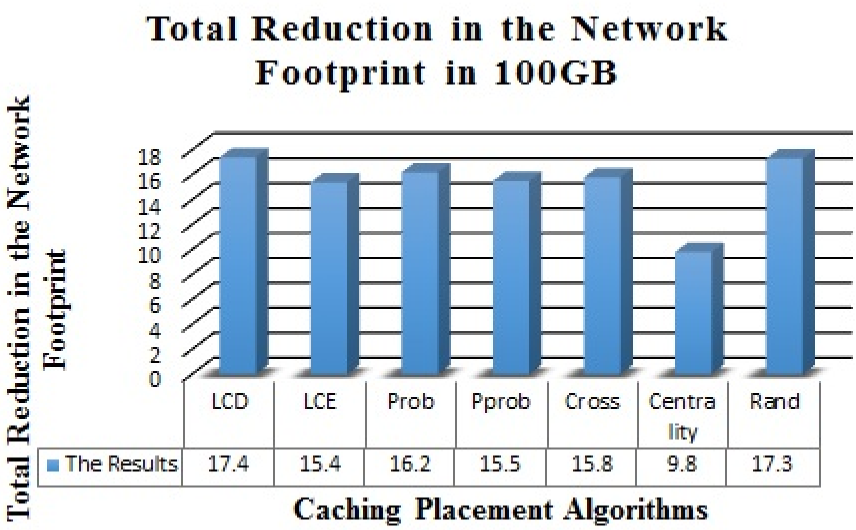


Figure 4.16: The Total Reduction in the Network Footprint for Caching Placement Algorithms in 100GB

Figure 4.17 shows a different attitude of the algorithms. In this case, Cross algorithm gives the highest value for a reduction in the network footprint, to which both Rand and Pprob resulted in similar values of 25. However, the Centrality algorithm again was the lowest.

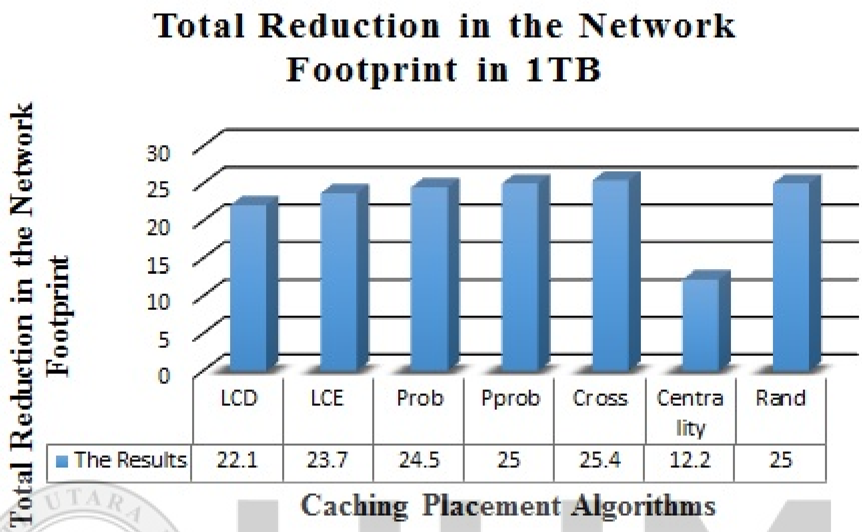


Figure 4.17: The Total Reduction in the Network Footprint for Caching Placement Algorithms in 1TB

Table 4.3: The Average Total Reduction in the Network Footprint in Four Cache Sizes

Cache Size	LCD	LCE	Prob	Pprob	Cross	Centrality	Rand
1GB	3.6	2.5	2.3	1.8	2.5	1.8	2.3
10GB	9.6	5.7	5.9	6.4	5.8	4.5	7.6
100GB	17.4	15.4	16.2	15.5	15.8	9.8	17.3
1TB	22.1	23.7	24.5	25	25.4	12.2	25
Average	13.175	11.825	12.225	12.175	12.375	7.075	13.05

Hence, we can generally conclude that from Figure 4.14 to Figure 4.17 that as cache size increases, total reduction in the network footprint also increases for all seven presented algorithms as shown in Figure 4.18 that elaborates the Table 4.3. Whereas, LCD and Cross algorithms give the highest reduction in the network footprint, while the Centrality algorithm gives the lowest reduction. However, Centrality algorithm is the worst because the content node may drop in the event of a high number of content delivery paths, which reasoned to the cache misses within a network [19]. However, we can see that LCD algorithm registers the best result with cache sizes (1GB, 10GB, and 100GB). Whenever, CROSS algorithm has the best result using the cache size of 1TB. In addition, Pprob algorithm produces the worst value with 1GB cache size. So, LCD algorithm again is the best value of Footprint reduction than the LCE and the Prob algorithms as the study [4, 83].

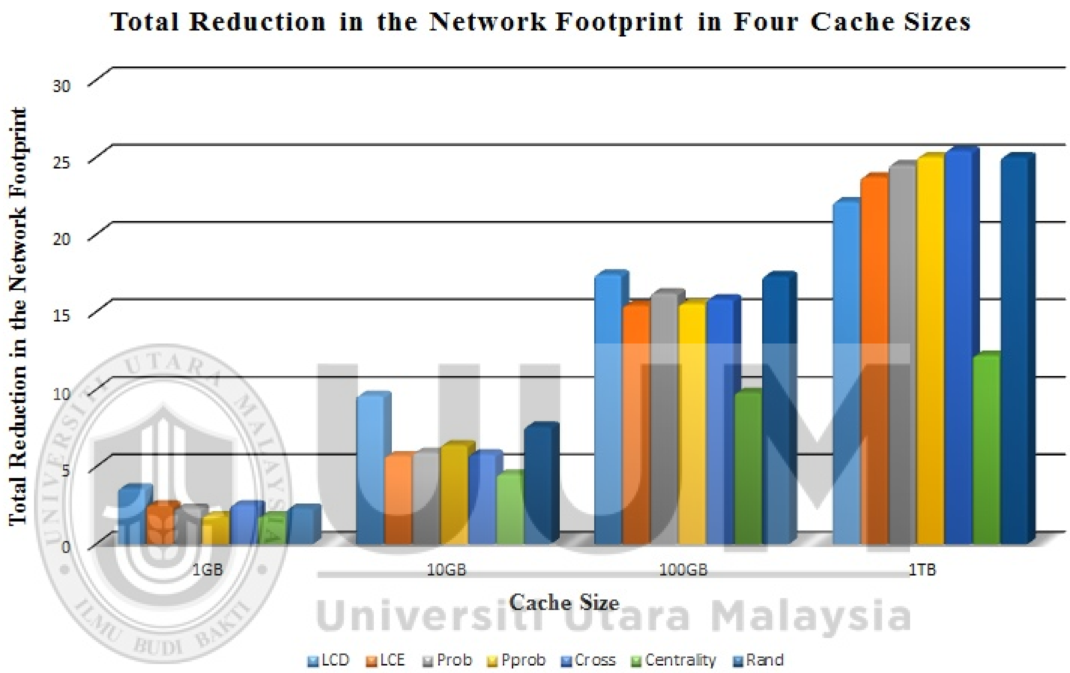


Figure 4.18: The Total Reduction in the Network Footprint for Caching Placement Algorithms in Four Cache Sizes

4.4.4 Reduction in the Server Load

Reduction in the server load on the origin servers is a very important performance metrics and [4]. It could be defined using the following equation:

$$1 - \frac{NumCachedReq}{V} \quad (4.6)$$

In the above equation *NumCachedReq* presents the number of requests that were served from any in-network cache. On the other hand, the variable *V* presents all video requests.

In the following diagrams, the values of the reduction in the server load for all seven cache placement algorithms for different cache sizes is presented. Figure 4.19 presents the results of the reduction in the server load expressed in % for all seven algorithms when the cache size is 1GB. It is obvious that the reduction in the server load is largest when LCD algorithm is used, and the Centrality algorithm gives the lowest reduction in the server load. As such, LCD algorithm gives the most favorable results because the higher reduction in the server load gives better overall performances.

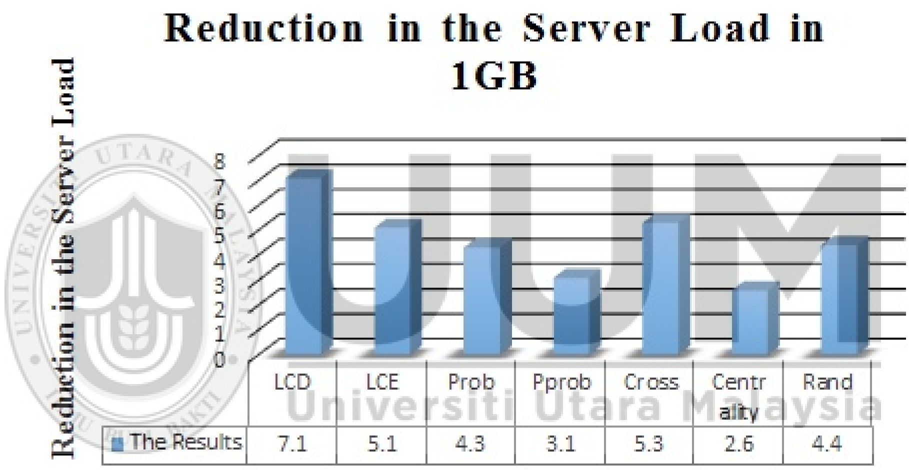


Figure 4.19: Reduction in Server Load for Caching Placement Algorithms in 1GB

Figure 4.20 shows the results of the reduction in the server load for all seven algorithms when the cache size is 10GB. Here we can see that the reduction in the server load of the Prop and Pprob is the same (16.5%) which considered being the highest. Meanwhile, other algorithms like Rand, Cross and LCE were found to provide a convenience results. Centrality algorithm continues to give the lowest results; in this case, its value for the reduction in the server load is lower than 7.2%.

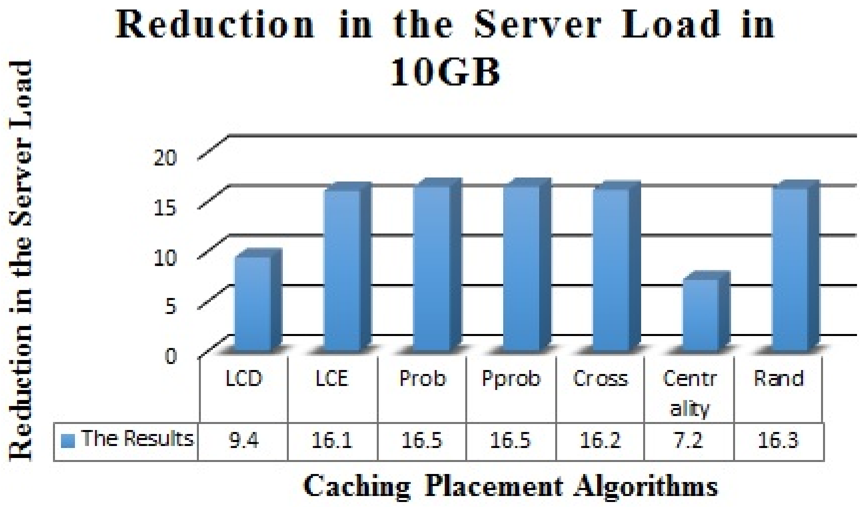


Figure 4.20: Reduction in Server Load for Caching Placement Algorithms in 10GB

Figure 4.21 presents the results of the reduction in the server load when the cache size is 100GB. It is clearly shown that LCD algorithm gives the highest value of around 28.4% of the reduction in the server load, while the Centrality algorithm provides the lowest performance of 15.2%.

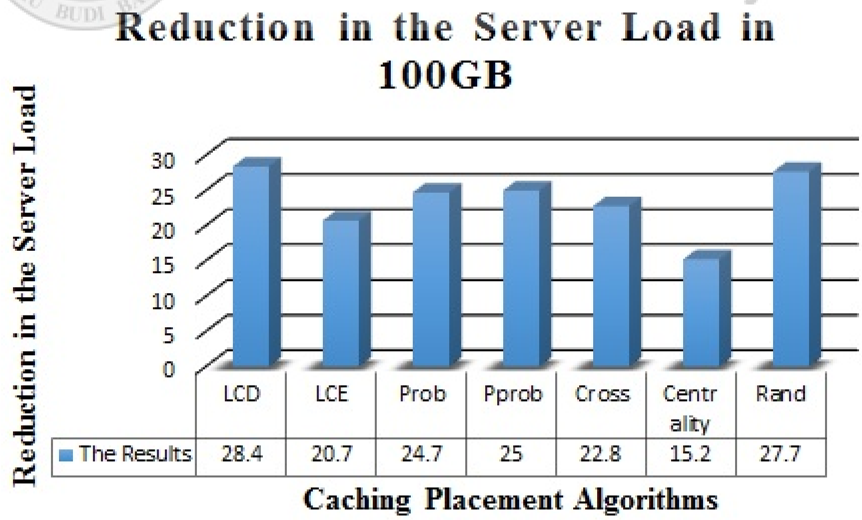


Figure 4.21: Reduction in Server Load for Caching Placement Algorithms in 100GB

Figure 4.22 shows a reduction in the server load results for all seven algorithms when the cache size is 1TB. From the results, it is obvious that Pprob algorithm has the greatest value of the reduction in the server load of around 45.3% followed by Rand algorithm (43.8%). Centrality algorithm has the lowest value of less than 19.2%.

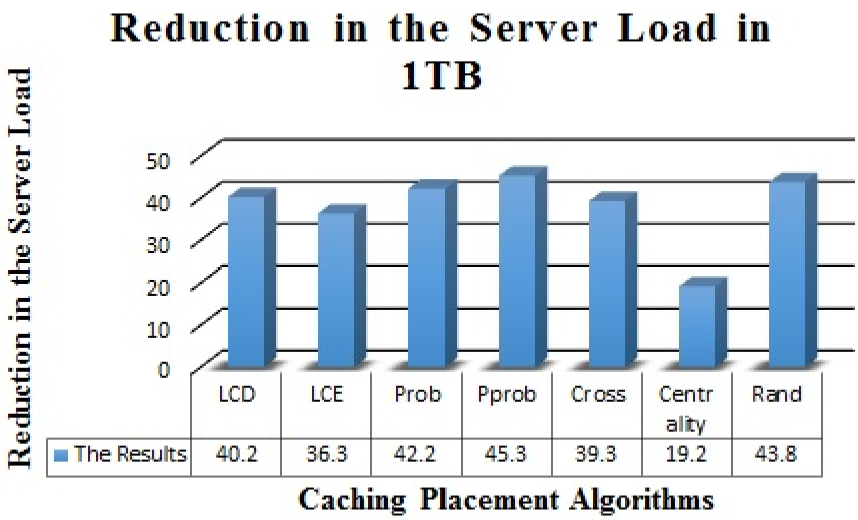


Figure 4.22: Reduction in Server Load for Caching Placement Algorithms in 1TB

From the presented diagrams from Figure 4.19 to Figure 4.22, we can conclude that the reduction in the server load is increased when the cache size is increasing for all seven algorithms.

Table 4.4: Average Reduction in the Server Load in Four Cache Sizes

Cache Size	LCD	LCE	Prob	Pprob	Cross	Centrality	Rand
1GB	7.1	5.1	4.3	3.1	5.3	2.6	4.4
10GB	9.4	16.1	16.5	16.5	16.2	7.2	16.3
100GB	28.4	20.7	24.7	25	22.8	15.2	27.7
1TB	40.2	36.3	42.2	45.3	39.3	19.2	43.8
Average	21.275	19.55	21.925	22.475	20.9	11.05	23.05

However, in the Figure 4.23 below, we can see that Centrality almost is the worst performance of reduction in server load with four cache sizes that is very clear in Table 4.4. While Prob and Pprob algorithms give the same result with cache size 10GB and they are the best using 10GB. However, Pprob is the best algorithm when the cache sizes are 10GB and 1TB. In addition, the reduction in server load of LCD algorithm is the best result when using cache size with 1GB, and 100GB.

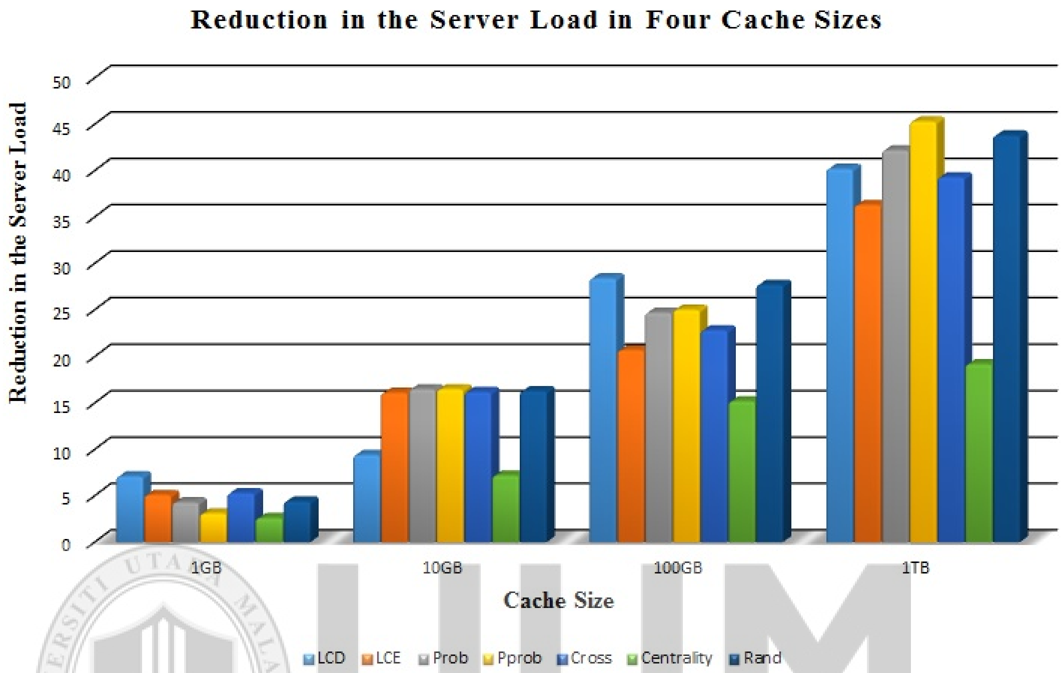


Figure 4.23: The Reduction in server Load for Caching Placement Algorithms in Four Cache Sizes.

Furthermore, the Rand algorithm saves the second level algorithm in a reduction in server load. Whereas it has the second level with using cache size in 10GB, 100GB, and 1TB because the random scheme deletes a content for each request to the server, so only one content is replaced with the arrival of a new content [84]. Therefore, the reduction in server load is much increased. Moreover, it was set to place a copy of content file on a randomly designated node, whereas its reference packet is processed through the network to build up the potential field for node caching.

4.5 Summary

In this work, the researcher has evaluated seven cache placement algorithms, LCD, LCE, Prob, Pprob, Cross, Centrality, and Rand on a CloudTV network in order to provide depth. The researcher further examined the performance metrics of delay time, average cache hit ratio, total reduction in the network footprint, and reduction in the server load for the seven cache placement algorithms above. Additionally, the researcher compared the results of these algorithms based on cache size from 1GB to 1 TB.

CHAPTER FIVE
CONCLUSION AND FUTURE WORKS

This dissertation aims to evaluate the widely used NDN cache placement algorithms. The performances of the placement algorithms are evaluated using NS3, and CloudTV content as VoD. This chapter provides the conclusion of the research work. It starts with Section 5.1, whereas the research findings. Secondly, Section 5.2 focuses on the contributions of this research. Then, the limitation and the challenges of the research is explained in Section 5.3. Finally, Section 5.4 suggests some future studies.

5.1 Summary of the Research

As cloud environments become more heavily used, provisioning the underlying network becomes more important. Previous approaches to deal with this problem involve changing the network infrastructure, for instance by imposing a particular topology or creating a new routing protocol. While these techniques are interesting and successful in their own right, the researcher in this study asks what has the best results of the four performance metrics of the seven cache placement algorithms. That is, in order to firstly enhanced cloud network performance without necessarily involving the provider, and secondly, that different algorithms can be immediately applicable to today's cloud networks.

Hence, the researcher in this study evaluated the several widely used placement algorithms named LCD, LCE, Prob, Pprob, Cross, Centrality, and Rand on a CloudTV network using NS3. This was mainly conducted to determine the current difficulties associated with the current content placement algorithms. The researcher designed the topology in Internet2 settings, which revealed to offer a fractional storage scheme and use appropriate network codes to transform the nodes over networks.

On the other hand, the researcher customizes the topology selection to fit the on a CloudTV, which transforms the nodes into a sequence destination. The performance metrics analyzing showed that when the cache size is increased from 1 GB to 1 TB, this leads to increase the average cache hit ratio, the reduction in network footprint, and the reduction in server load, whereas, thus increasing gives the best results. Whenever, the delay becomes lower when the cache size increases that is good.

Based on the obtained results, it can be concluded that the Prob, Pprob, and LCE algorithms the best results of time delay with cache sizes (1TB, 100GB, 10GB, 1GB) respectively. Wherever, the Prob algorithm has the best delay time with 1TB and 100GB. However, Rand algorithm gives the best result of the average cache hit ratio in 1TB and 100GB. While, LCD gives the best results with 10GB and 1GB. Whenever, Prob is the best in average hit ratio with 1TB only.

In addition, it can conclude that LCD algorithm is the preferable algorithm in the total reduction in network footprint when the cache sizes are 100GB, 10GB, and 1GB. While cross algorithm produces better results with 1TB only. Furthermore, the reduction in server load results that LCD gives the highest value with 100GB and 1GB. However, Pprob has the highest value with 1TB and 10GB. Also, Prob obtains the preferable value of the reduction in server load with 10GB.

Finally, we can conclude that Centrality is the worst cache placement algorithm, because it has the worst value in the average cache hit ratio, the total reduction in network footprint, and the reduction in server load with all four cache sizes (1GB, 10GB, 100GB, 1TB). Also, it gives the worst performance in delay time with 1GB cache size. Centrality algorithm has the worst results because the content node may drop in the event of a high number of content delivery paths, which reasoned to the cache misses within a network.

5.2 Research Contributions

This research intends to contribute the Name data Network (NDN) algorithms of VoD services through the caching placement algorithms of online videos and comparing the performance of these algorithms in terms of performance metrics. The benefit of VoD contributes to the aspect of science and technology, also in other domains such as sociology, arts and government. The countless benefits from using the Internet, especially the educational value, schools have now acquired the computer systems with telecommunication tools to have access to the Internet. These videos ensure users to play the videos as the playback rate without any quality degradation while potentially reducing the publisher's bandwidth costs of video playback.

In addition, the study mentioned the academically, socially and morally driven benefits and implications: curriculum development of secondary school students, school policies on information retrieval, designers of search engines, and most importantly the teachers responsible for using the Internet as an educational tool in class. The impact of videos in education is far-reaching as it develops interactivity with the content, which further helps learners in their cognitive development. VoD in real time enables learners to engage more with learning and more specifically with the flow of knowledge transfer and memory.

The current study will focus on resolving the issues of online video traffic affecting both providers and users of the online video host. There would be a lot of revenue for the providers of service by Internet and indirectly the host of online video once the delay in responding to the request of VoD workload is reduced.

5.3 Research Limitation

The VoD services frequently face a number of challenges such as heavy video traffic, delay in VoD performance, constraints in content distribution, and so on, this happened in case of large-scale content-centric networks. For instance, with the rapid increase in user requests for online videos, the storage capacity also gets increased which further makes replication of the entire storage data almost invisible. Moreover, constant responses to the user requests for online videos increase video traffic to an extent that delays the performance of the VoD service and more particularly it affects content caching in the router. At the users' end, this further leads to decrease in service quality and high access delays due to long download times.

This task is particularly challenging in cellular networks due to the inherent variability in signal strength, interference, noise, and user mobility causing the bandwidth to vary widely over time Therefore, to ensure a fully utilized network, it is essential that delay related challenges are addressed. Those researchers explained the types of caching that promote reducing network delay while sending and receiving data which reasoned to the storage of data in a certain node when network faces sudden disruptions.

However, the prompt response poses a serious challenge as different request often lead to delays in VoD workload, which further affects the content cache of the router. Giving a prompt reply to the user requests of online videos would serve as activation for the performance of the network. Therefore, more responses to the different user requests of online videos would lead to delay in network performance.

5.4 Future works

The researcher proposes a number of aspects to be talked by future studies, these are:

• One potential direction for future studies is to alter the current examination of placement algorithms to include multiple congestion metrics for resource provisioning of Video on Demand services. This includes investigating how much storage and bandwidth resources is required in order to offer a certain packet?

• Meanwhile, future studies can examine the theoretical dimensions of CloudTV by considering other video applications including real time streaming, video conferencing, video gaming and etc.

• Finally, future studies can examine the system size by pursuing system development at a larger scale with noting the practical aspects that affect the network bandwidth fluctuation and transmission delay, along with the cache server performance such as disk read/write speed and system failure.

REFERENCES

[1] I. Psaras, R. G. Clegg, R. Landa, W. K. Chai, and G. Pavlou, "Modelling and evaluation of ccn-caching trees," in *NETWORKING 2011*, pp. 78–91, Springer, 2011.

[2] K. M. Zaini, A. M. Habbal, F. Azzali, S. Hassan, and M. Rizal, "An interaction between congestion-control based transport protocols and manet routing protocols," *Journal of Computer Science*, vol. 8, no. 4, p. 468, 2012.

[3] 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*, pp. 55–60, ACM, 2012.

[4] Y. Sun, S. K. Fayaz, Y. Guo, V. Sekar, Y. Jin, M. A. Kaafar, and S. Uhlig, "Trace-driven analysis of icn caching algorithms on video-on-demand workloads," in *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, pp. 363–376, ACM, 2014.

[5] C. Yi, J. Abraham, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "On the role of routing in named data networking," in *Proceedings of the 1st international conference on Information-centric networking*, pp. 27–36, ACM, 2014.

[6] X. Liu, F. Dobrian, H. Milner, J. Jiang, V. Sekar, I. Stoica, and H. Zhang, "A case for a coordinated internet video control plane," in *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, pp. 359–370, ACM, 2012.

[7] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," in *Proceedings of the 2014 ACM conference on SIGCOMM*, pp. 187–198, ACM, 2014.

[8] J. Liu, A. Panda, A. Singla, B. Godfrey, M. Schapira, and S. Shenker, "Ensuring connectivity via data plane mechanisms.," in *NSDI*, pp. 113–126, 2013.

[9] C. Bian, Z. Zhu, A. Afanasyev, E. Uzun, and L. Zhang, "Deploying key management on ndn testbed," *UCLA, Peking University and PARC, Tech. Rep*, 2013.

[10] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos, *et al.*, "Named data networking (ndn) project," *Relatorio Tecnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 2010.

[11] G. Xylomenos, C. N. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, G. C. Polyzos, *et al.*, "A survey of information-centric networking research," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 2, pp. 1024–1049, 2014.

[12] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, “A survey of energy-efficient caching in information-centric networking,” *Communications Magazine, IEEE*, vol. 52, no. 11, pp. 122–129, 2014.

[13] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, “A case for stateful forwarding plane,” *Computer Communications*, vol. 36, no. 7, pp. 779–791, 2013.

[14] A. Balachandran, V. Sekar, A. Akella, and S. Seshan, “Analyzing the potential benefits of cdn augmentation strategies for internet video workloads,” in *Proceedings of the 2013 conference on Internet measurement conference*, pp. 43–56, ACM, 2013.

[15] X. Yin, V. Sekar, and B. Sinopoli, “Toward a principled framework to design dynamic adaptive streaming algorithms over http,” in *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, p. 9, ACM, 2014.

[16] N. Laoutaris, H. Che, and I. Stavrakakis, “The lcd interconnection of lru caches and its analysis,” *Performance Evaluation*, vol. 63, no. 7, pp. 609–634, 2006.

[17] J. Li, H. Wu, B. Liu, and J. Lu, “Effective caching schemes for minimizing inter-isp traffic in named data networking,” in *Parallel and Distributed Systems (ICPADS), 2012 IEEE 18th International Conference on*, pp. 580–587, IEEE, 2012.

[18] W. Wang, Y. Sun, Y. Guo, D. Kaafar, J. Jin, J. Li, and Z. Li, “Crcache: exploiting the correlation between content popularity and network topology information for icn caching,” in *Communications (ICC), 2014 IEEE International Conference on*, pp. 3191–3196, IEEE, 2014.

[19] W. KOONG CHAI, H. DILIANG, I. Psaras, and G. Pavlou, “Cache less for more in information-centric networks (extended version),” *Computer communications*, vol. 36, no. 7, pp. 758–770, 2013.

[20] Z. Zhu and A. Afanasyev, “Let’s chronosync: Decentralized dataset state synchronization in named data networking,” in *Network Protocols (ICNP), 2013 21st IEEE International Conference on*, pp. 1–10, IEEE, 2013.

[21] Z. Ming, M. Xu, and D. Wang, “Age-based cooperative caching in information-centric networks,” in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pp. 268–273, IEEE, 2012.

[22] C. Huang, J. Li, and K. W. Ross, “Can internet video-on-demand be profitable?,” *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 133–144, 2007.

[23] U. O. Christopher and E. Maria-Gorretti, “Availability and the use of computer and internet by secondary school students in benin city, nigeria,” *International Journal of Library and Information Science*, vol. 4, no. 2, pp. 16–23, 2012.

[24] A. D. Greenberg and J. Zanetis, “The impact of broadcast and streaming video in education,” *Cisco: Wainhouse Research*, 2012.

[25] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. K. Ramakrishnan, "Optimal content placement for a large-scale vod system," in *Proceedings of the 6th International COnference*, p. 4, ACM, 2010.

[26] A.-M. K. Pathan and R. Buyya, "A taxonomy and survey of content delivery networks," *Grid Computing and Distributed Systems Laboratory, University of Melbourne, Technical Report*, 2007.

[27] M. Gritter and D. R. Cheriton, "An architecture for content routing support in the internet.," in *USITS*, vol. 1, pp. 4–4, 2001.

[28] 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*, pp. 1–12, ACM, 2009.

[29] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *ACM SIGCOMM Computer Communication Review*, vol. 37, pp. 181–192, ACM, 2007.

[30] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, "Dos and ddos in named data networking," in *Computer Communications and Networks (ICCCN), 2013 22nd International Conference on*, pp. 1–7, IEEE, 2013.

[31] J. Li, H. Wu, B. Liu, and J. Lu, "Effective caching schemes for minimizing inter-isp traffic in named data networking," in *Parallel and Distributed Systems (ICPADS), 2012 IEEE 18th International Conference on*, pp. 580–587, IEEE, 2012.

[32] M. Ambrosin, C. Busold, M. Conti, A.-R. Sadeghi, and M. Schunter, "Updaticator: Updating billions of devices by an efficient, scalable and secure software update distribution over untrusted cache-enabled networks," in *Computer Security-ESORICS 2014*, pp. 76–93, Springer, 2014.

[33] D. Turner, K. Levchenko, S. Savage, and A. C. Snoeren, "A comparison of syslog and is-is for network failure analysis," in *Proceedings of the 2013 conference on Internet measurement conference*, pp. 433–440, ACM, 2013.

[34] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, Y. Ganjali, and C. Diot, "Characterization of failures in an operational ip backbone network," *IEEE/ACM Transactions on Networking (TON)*, vol. 16, no. 4, pp. 749–762, 2008.

[35] J. Erman, A. Gerber, K. Ramadrishnan, S. Sen, and O. Spatscheck, "Over the top video: the gorilla in cellular networks," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pp. 127–136, ACM, 2011.

[36] A. Suki and C. M. Arif, *Slight-Delay Shaped Variable Bit Rate (SD-SVBR) Technique for Video Transmission*. PhD thesis, Universiti Utara Malaysia, 2011.

[37] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang, "Interest flooding attack and countermeasures in named data networking," in *IFIP Networking Conference, 2013*, pp. 1–9, IEEE, 2013.

[38] A. Galanopoulos, G. Iosifidis, A. Argyriou, and L. Tassiulas, "Green video delivery in lte-based heterogeneous cellular networks," *in Proceedings of the 19th annual international conference on Mobile computing & networking*, pp. 389–400., 2013.

[39] C. V. N. Index, "Forecast and methodology, 2012–2017. cisco white paper, may 29, 2013," 2014.

[40] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, and S. Shenker, "Less pain, most of the gain: Incrementally deployable icn," in *ACM SIGCOMM Computer Communication Review*, vol. 43, pp. 147–158, ACM, 2013.

[41] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker, "Naming in content-oriented architectures," in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, pp. 1–6, ACM, 2011.

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

[43] M. Draxler 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*, pp. 581–587, IEEE, 2012.

[44] H. Nam, K.-H. Kim, J. Y. Kim, and H. Schulzrinne, "Towards qoe-aware video streaming using sdn," in *Global Communications Conference (GLOBECOM), 2014 IEEE*, pp. 1317–1322, IEEE, 2014.

[45] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pp. 97–108, ACM, 2012.

[46] R. K. Mok, X. Luo, E. W. Chan, and R. K. Chang, "Qdash: a qoe-aware dash system," in *Proceedings of the 3rd Multimedia Systems Conference*, pp. 11–22, ACM, 2012.

[47] R. Margolies, A. Sridharan, V. Aggarwal, R. Jana, N. Shankaranarayanan, V. Vaishampayan, G. Zussman, *et al.*, "Exploiting mobility in proportional fair cellular scheduling: Measurements and algorithms," in *INFOCOM, 2014 Proceedings IEEE*, pp. 1339–1347, IEEE, 2014.

[48] Q. Xu, S. Mehrotra, Z. Mao, and J. Li, "Proteus: network performance forecast for real-time, interactive mobile applications," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pp. 347–360, ACM, 2013.

[49] M. Claeys, D. Tuncer, J. Famaey, M. Charalambides, S. Latre, F. De Turck, and G. Pavlou, "Towards multi-tenant cache management for isp networks," in *Networks and Communications (EuCNC), 2014 European Conference on*, pp. 1–5, IEEE, 2014.

[50] W. Rao, L. Chen, A. W.-C. Fu, and G. Wang, "Optimal resource placement in structured peer-to-peer networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 21, no. 7, pp. 1011–1026, 2010.

[51] L. Wang, S. Bayhan, and J. Kangasharju, "Cooperation policies for efficient in-network caching," in *ACM SIGCOMM Computer Communication Review*, vol. 43, pp. 533–534, ACM, 2013.

[52] S. P. Shariatpanahi, S. A. Motahari, and B. H. Khalaj, "Multi-server coded caching," *arXiv preprint arXiv:1503.00265*, 2015.

[53] 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*, pp. 66–75, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

[54] N. Laoutaris, S. Syntila, and I. Stavrakakis, "Meta algorithms for hierarchical web caches," in *Performance, Computing, and Communications, 2004 IEEE International Conference on*, pp. 445–452, IEEE, 2004.

[55] K.-Y. Wong, "Web cache replacement policies: a pragmatic approach," *Network, IEEE*, vol. 20, no. 1, pp. 28–34, 2006.

[56] S. Saha, A. Lukyanenko, and A. Ylä-Jääski, "Efficient cache availability management in information-centric networks," *Computer Networks*, vol. 84, pp. 32–45, 2015.

[57] L. Yin and G. Cao, "Supporting cooperative caching in ad hoc networks," *Mobile Computing, IEEE Transactions on*, vol. 5, no. 1, pp. 77–89, 2006.

[58] R. Ravindran, S. Lo, X. Zhang, and G. Wang, "Supporting seamless mobility in named data networking," in *Communications (ICC), 2012 IEEE International Conference on*, pp. 5854–5869, IEEE, 2012.

[59] 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*, pp. 15–26, ACM, 2012.

[60] C. Cabral, C. E. Rothenberg, and M. F. Magalhaes, "Mini-ccnx: fast prototyping for named data networking," in *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, pp. 33–34, ACM, 2013.

[61] B. Han, X. Wang, N. Choi, T. Kwon, and Y. Choi, "Amvs-ndn: Adaptive mobile video streaming and sharing in wireless named data networking," in *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, pp. 375–380, IEEE, 2013.

[62] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Pending interest table sizing in named data networking," in *Proceedings of the 2nd International Conference on Information-Centric Networking*, pp. 49–58, ACM, 2015.

[63] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive forwarding in named data networking," *ACM SIGCOMM computer communication review*, vol. 42, no. 3, pp. 62–67, 2012.

[64] J. Tang, Z. Zhang, Y. Liu, and H. Zhang, "Identifying interest flooding in named data networking," in *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, pp. 306–310, IEEE, 2013.

[65] M. U. Haque, K. Pawlikowski, A. Willig, and L. Bischofs, "Performance analysis of blind routing algorithms over content centric networking architecture," in *Computer and Communication Engineering (ICCCE), 2012 International Conference on*, pp. 922–927, IEEE, 2012.

[66] G. Tyson, J. Bigham, and E. Bodanese, "Towards an information-centric delay-tolerant network," in *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, pp. 387–392, IEEE, 2013.

[67] G. A. Churchill, T. J. Brown, and T. A. Suter, "Basic marketing research," 2004.

[68] A. Bryman and E. Bell, *Business research methods*. Oxford university press, 2015.

[69] D. A. De Vaus and D. de Vaus, *Research design in social research*. Sage, 2001.

[70] U. Sekaran, *Research methods for business: A skill building approach*. John Wiley & Sons, 2006.

[71] G. Lancaster, *Research methods in management*. Routledge, 2007.

[72] U. Sekaran, "Research methods: A skill building approach. john wiley & sons," 2011.

[73] R. Ahmed, M. F. Bari, S. R. Chowdhury, M. G. Rabbani, R. Boutaba, and B. Mathieu, "alpha route: A name based routing scheme for information centric networks," in *INFOCOM, 2013 Proceedings IEEE*, pp. 90–94, IEEE, 2013.

[74] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and Tools for Network Simulation*, pp. 15–34, Springer, 2010.

[75] T. D. Assefa, "Qos performance of lte networks with network coding," 2015.

[76] W. So, A. Narayanan, and D. Oran, "Named data networking on a router: Fast and dos-resistant forwarding with hash tables," in *Proceedings of the ninth ACM/IEEE symposium on Architectures for networking and communications systems*, pp. 215–226, IEEE Press, 2013.

[77] L. Saino, I. Psaras, and G. Pavlou, "Hash-routing schemes for information centric networking," in *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, pp. 27–32, ACM, 2013.

[78] B. Stockebrand, "Quality of service (qos)," *IPv6 in Practice: A Unixers Guide to the Next Generation Internet*, pp. 327–331, 2007.

[79] R. Asokan and A. Natarajan, "An approach for reducing the end-to-end delay and increasing network lifetime in mobile adhoc networks," *Int J Inf Technol*, vol. 4, no. 2, pp. 121–127, 2008.

[80] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 26–36, 2012.

[81] I. Psaras, W. K. Chai, and G. Pavlou, "In-network cache management and resource allocation for information-centric networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 11, pp. 2920–2931, 2014.

[82] G. Zhang, B. Tang, X. Wang, and Y. Wu, "An optimal cache placement strategy based on content popularity in content centric network," *JOICE*, pp. 2759–2967, 2014.

[83] G. Zhang, B. Tang, P. Wang, Y. Wu, and X. Zhang, "Performance assessment of cache strategies in content centric network," *Information Technology Journal*, vol. 12, no. 23, p. 7083, 2013.

[84] J. Sung, J.-K. K. Rhee, and S. Jung, "Lightweight caching strategy for wireless content delivery networks," *IEICE Communications Express*, vol. 3, no. 4, pp. 150–155, 2014.