CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

Resource Efficient Multi-dimensional Cache Management Strategies in Content-Centric Networks

by

Sheneela Naz

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the

Faculty of Computing

Department of Computer Science

2018

Resource Efficient Multi-dimensional Cache Management Strategies in Content-Centric Networks

By

Sheneela Naz

(PC103011)

Dr. Feng Xia, Professor

Dalian University of Technology (DUT), P.R. of China

Dr. Faisal Tariq

Queens Marry University of London, London, UK

Dr. Amir Qayyum

(Supervisor)

Dr. Muhammad Abdul Qadir

(Dean, Faculty of Computing)

DEPARTMENT OF COMPUTER SCIENCE

CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY

ISLAMABAD

2018

Dedicated

to

My Beloved

Parents

CAPITAL UNIVERSITY OF SCIENCE & TECHNOLOGY
ISLAMABAD

Expressway, Kahuta Road, Zone-V, Islamabad
Phone:+92-51-111-555-666 Fax: +92-51-4486705
Email: info@cust.edu.pk Website: https://www.cust.edu.pk

CERTIFICATE OF APPROVAL

This is to certify that the research work presented in the thesis, entitled "**Resource Efficient Multi-dimensional Cache Management Strategy in Content Centric Networks**" was conducted under the supervision of **Dr. Amir Qayyum**. No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the **Department of Computer Science, Capital University of Science and Technology** in partial fulfillment of the requirements for the degree of Doctor in Philosophy in the field of **Computer Science.** The open defence of the thesis was conducted on **05 March, 2018**.

Student Name :	Ms. Sheneela Naz (PC103011)	Sheneela

The Examination Committee unanimously agrees to award PhD degree in the mentioned field.

Examination Committee :

(a) External Examiner 1:	Dr. Nadeem Javaid, Associate Professor CIIT, Islamabad	[illegible]
(b) External Examiner 2:	Dr. Muhammad Usman, Assistant Professor QAU, Islamabad	[illegible]
(c) Internal Examiner :	Dr. Muhammad Arshad Islam, Assistant Professor, CUST, Islamabad	Arshad
Supervisor Name :	Dr. Amir Qayyum, Professor, CUST, Islamabad	[illegible]
Name of HoD :	Dr. Nayyer Masood, Professor, CUST, Islamabad	[illegible]
Name of Dean :	Dr. Muhammad Abdul Qadir, Professor, CUST, Islamabad	[illegible]

PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled "**Resource Efficient Multi-dimensional Cache Management Strategy in Content Centric Networks**" is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/ cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of PhD Degree, the University reserves the right to withdraw/ revoke my PhD degree and that HEC and the University have the right to publish my name on the HEC/ University Website on which names of students are placed who submitted plagiarized thesis.

Sheneela

(Ms. Sheneela Naz)
Registration No. PC103011

Dated: 05 March, 2018

AUTHOR'S DECLARATION

I, **Ms. Sheneela Naz (Registration No. PC103011)**, hereby state that my PhD thesis titled, '**Resource Efficient Multi-dimensional Cache Management Strategy in Content Centric Networks**' is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/ world.

At any time, if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my PhD Degree.

Sheneela

(Ms. Sheneela Naz)

Dated: 05 March, 2018 Registration No : PC103011

List of Publications

It is certified that following publication(s) have been made out of the research work that has been carried out for this thesis:-

1. **Sheneela Naz**, Rao Naveed Bin Rais, and Amir Qayyum, “A Resource Efficient Multi-dimensional Cache Management Strategy in Content Centric Networks”, *Journal of Computational and Theoretical Nanoscience*, Vol. 14, 116, 2017.
2. **Sheneela Naz**, Rao Naveed Bin Rais, Peer Azmat Shah, Sadaf Yasmin and Amir Qayyum. A Dynamic Caching Strategy for CCN-based MANETs. The International Elsevier Computer Networks (COMNET), 2018.

Sheneela Naz
(Registration No. PC103011)

Acknowledgements

All Praises be to ALLAH Almighty who enabled us to complete this task successfully and our utmost respect to His last Prophet (P.B.U.H.).

I would like to sincerely thank my PhD supervisor, Dr. Amir Qayyum, for his guidance, encouragement and support throughout this study. I would also like to thank Dr. Rao Naveed Bin Rais who has been co-supervising my PhD work. His invaluable concern and guidance has helped me a lot in making the progress in my research. I would also like to mention the vital contribution I have received from Sadaf Yasmin, Peer Azmat Shah and Saira Gillani. I would like to thank the people from the CoReNeT. Their assistance was always effective and timely.

Finally, I thank my family who supporting me and encouraging me with their best wishes.

Abstract

CCN promises to solve many problems related to traditional network architecture with its receiver-driven, secure and simplistic model. Transparent and ubiquitous in-network caching, mobility management are two active research topics within CCN domain, and we intend to address both in this thesis. This document mainly consists of our work on improving in-network cache management in static and dynamic network environments.

In-network cache management in Content-Centric Networking (CCN) has received significant interest from research community in recent years. On the positive aspect, it helps in increasing content availability and quality-of-experience (QoE) by reducing end-to-end delays, and reduces server load. On the other hand, opting the default approach, that is, store a content at every node on the delivery path is not resource-efficient as it introduces high cache redundancy. A multitude of schemes are proposed to increase efficiency of the network, which aim to reduce redundancy by selecting a small fraction of nodes on returning path for storing contents. This selection is generally based on taking into account knowledge about the network or the content itself. However, in many schemes node's utility is generally determined by analyzing only one concept at a time. For example, some schemes store contents based on popularity, while others select nodes employing only the information about network topology etc. Considering only a single aspect while taking a decision may limit the scope of the cache management scheme to be deployed in diverse scenarios, and might result in sub-optimal performance when the environment is changed. Hence, there is a need to devise an efficient caching mechanism that can dynamically adjust and adapt itself to any environment. With this in mind, we propose an adaptive caching strategy, named as Multi-Attribute Caching Strategy (MACS) based on multi-parameters for CCN. MACS attempts to overcome inefficient cache utilization by intelligently selecting caching locations along the content delivery path. Simulation results show that MACS reduces cache load at each node while providing comparable delay and better cache hit rate using both synthetic and real network topologies.

In second part of the thesis, we analyze the aspects of content-centric caching in Mobile Ad-hoc Networks (MANET). We propose a caching framework that dynamically adapt the caching decision of each content and relocate the replica if the old cached node moves to another location. We envision that such cache relocation mechanism to stimulate cooperation among nodes have potential to help with practical deployments in real scenarios.

Contents

List of Figures

List of Tables

Abbreviations

ICN	Information-Centric Networking
CCN	Content-Centric Networking
CDN	Content Delivery Networks
MPC	Popularity-Based Caching Strategy
LCE	Leave Copy Everywhere
LCD	Leave Copy Down
RCone	Random Copy One
MCD	Move Copy Down

Symbols

p	Number of parameters
C	Number of contents
R	Number of routers
$\hbar_{(r)}$	Hop count/distance
$\Omega_{(s,r)}$	Distance or number of hops from the content provider s to current content router r
$\Omega_{(k,s)}$	Total distance or the number of hops from the client k to content provider s
$\rho_{r(c)}$	Content popularity
$\partial_{(r)}$	Degree score of node r
$d_{(r)}$	Node Degree
$d_{max}(s,k)$	Maximum degree from the content provider s to a client k
$A_{cs(r)}$	Available cache space at router r
$CD_{Score}(c)$	Composite cache decision score for content C at router r
w_i	Associated weight of parameter i
CR	Content router
μ	Threshold value
$N_{(freshness)}$	Current content freshness of attached to the upstream router
$Current_{(time)}$	Current time
Δw	Change in weight
l	Learning rate
η	Momentum parameter

Chapter 1

Introduction

1.1 Background

During the past decade, user requirements and emerging applications have challenged the traditional communication mechanisms to a great deal. Besides, there has been a tremendous growth in the number of users and applications [1], as shown in Figure. 1.1. In most of the emerging applications such as content sharing and video-on-demand etc. [2–4], users are more interested in acquiring a particular content, rather than the location from where the content is coming, and most of the time, are oblivious to the source identification as well. Figure. 1.2 shows the trends that are changed from host-oriented communication to data-oriented communication [1].

Indeed, in content sharing mechanisms, 'what is exchanged' is becoming more important than 'who is exchanging' it. As a result, the focus of connectivity is moving from interconnecting machines to interconnecting information. Content distribution between two endpoints from content providers to consumers involves many technologies e.g. Content Delivery Networks (CDN), Peer-to-peer (P2P) Networks, etc. These technologies build an information-centric service model over

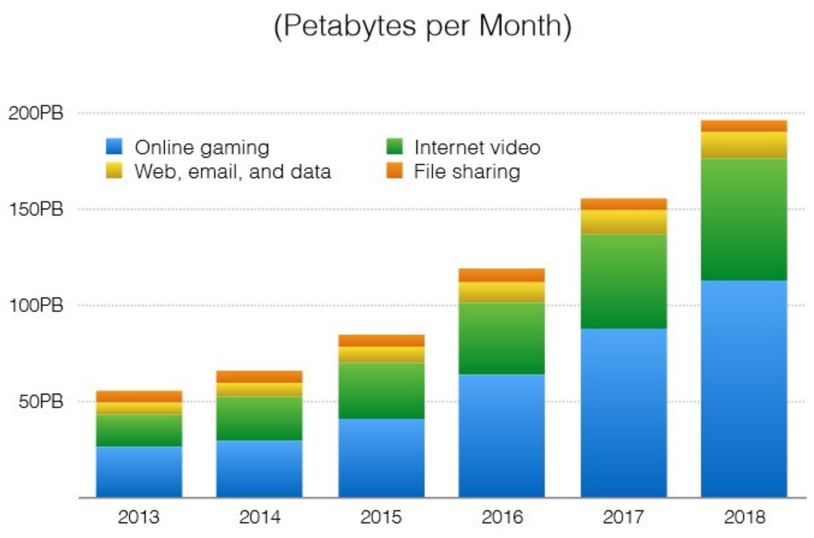


FIGURE 1.1: Traffic growth

a network infrastructure which was intended to support host-to-host communications. This information-centric service model currently does not take full advantage of resources along the path from consumer to provider. The premise is that content delivery can be improved by including network storage. Network storage is an opportunity to increase service availability and to reduce the network traffic and server load. Figure. 1.3 illustrates how network storage has been evolved.

P2P and CDN are network caching solutions that have been deployed to overcome the inherent limitations of the Internet in terms of user perceived Quality of Service (QoS) when accessing Web content [5]. However, CDN may experience sub-optimal performance due to 1) no control over the placement of the CDNs server, 2) lack of communication among different CDN servers and 3) CDNs only serve contents for a subset of applications due to the agreements with content providers [6]. P2P systems receive a lot of inter-ISP (Internet Service Provider inter-connections) traffic and are unstable in terms of content availability and download performance. In order to overcome these limitations and to support content retrievals efficiently, content-oriented networking architecture was proposed that is known as an Information-Centric Networking (i.e. ICN).

ICN is a communication paradigm where content names are disassociated from host addresses and named data is exchanged rather than sending data packets

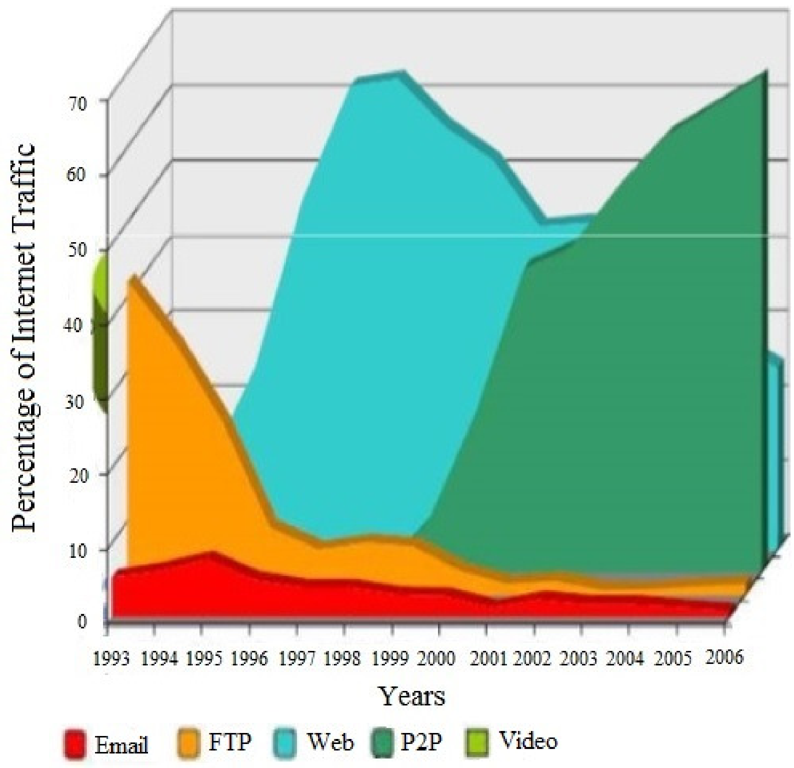


FIGURE 1.2: Trends change from host-oriented communication to data-oriented communication

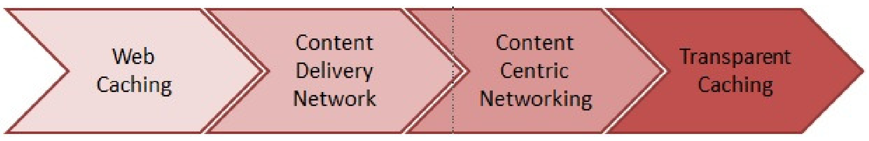


FIGURE 1.3: Network storage evolution

from source to destination [7]. Therefore, users are interested in *'what the content is'* rather *'where the content is located'*. The ICN paradigm exploits the concept of in-network caching in order to expedite content distribution and improve network resource utilization. As compared with web caching and CDN caching, the ICN caching is transparent, ubiquitous and has a fine-granularity [8, 9]. Examples of ICN architectures in the literature include Content-Centric Networking(CCN)/NDN [10], Data-Oriented Network Architecture (DONA) [11], Publish Subscribe Internet Routing Paradigm (PSIRP) [12] and the Network of Information (NetInf) [13] etc.

Ubiquitous in-network caching of content is a focal point of CCN architecture.

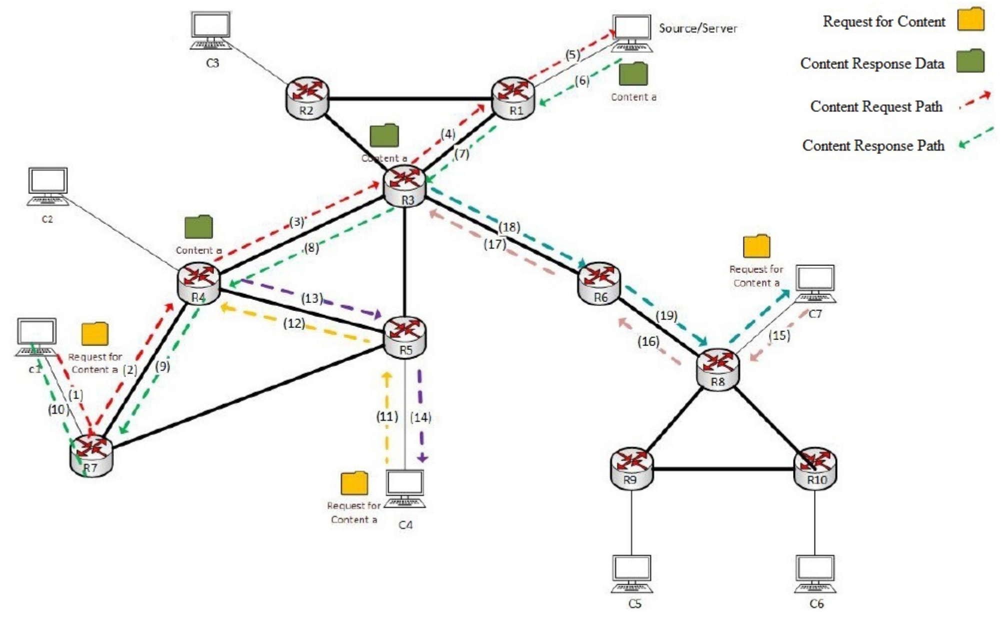


FIGURE 1.4: Example of in-network caching

CCN promises to solve many problems related to the traditional network architecture as it provides a receiver-driven, secure and simplistic model while inherently support multiple interfaces per node, and unicast, multicast and broadcast communication modes. CCN provides transparent and ubiquitous in-network caching, name-based routing [11] and content-level protection [14].

CCN utilizes the concept of in-network caching in order to enhance content availability in the network. As data packet traverses through the network, each content router on the downloading path can potentially cache the content, as per the default cache policy of the architecture. A consumer broadcasts its interest packet of content over all available connectivity in the network. Each node which receives the interest packet checks in its local cache and if having the requested content is found, then transmits the data packet to the consumer and does not forward the request any further. If the receiving node has no such content, then it forwards the request to the next content router hoping to find the content. The process of in-network caching is presented in Figure. 1.4.

As per the default cache policy of CCN, each content router on the downloading path can potentially cache the contents to answer future queries. This avoids the need to send future interest packets for the same content to the server, and thus results in reducing delay and server load thereby increasing overall user Quality of Experience (QoE). However, the default strategy is not resource-efficient as every content has to be unnecessarily cached on each content router (CR) even if the same content is available at the neighboring nodes in the vicinity.

There are some cache management challenges which are:

1. *Where to save*: Deciding on which routers on the path the content should be saved for an increased performance is a challenging issue. In most cases, nodes located at strategically important locations are selected to save a content to maximize reusability. Therefore Degree Centrality (DC)[15], Betweeness Centrality (BC)[16] and Closeness Centrality (CC)[17] are popular techniques to determine the feasibility of a node in terms of number of clients that it can serve.

2. *What to save*: What to save is also very important along with where to save. However, what to save is very much related to the present traffic pattern in the network. Saving a content near the region where it is most popular have the potential to save network bandwidth and reduce end-to-end delay experienced by the end users. Hence where to save and what to save are two most important questions which every scheme is required to address.

3. *How to manage resources*: Intermediate routers have finite cache space and may have high storage cost. Therefore, resource management is another challenge issue. Resource utilization is one of the important performance parameters that is use to evaluate in-network cache techniques. Along with avoiding congestion at a node, it is also important to efficiently utilize collective resources within a region. Therefore, most of the schemes try to reduce redundant copies of a content placed within close vicinity.

4. *How fresh the content is*: For how much duration the content should be stored is also very important e.g if we take the example of live cricket match which is also stored on intermediate routers while it is streaming. It is possible to receive maximum number of requests, but it will lose its popularity after a day

or even after few hours. Hence, it also required to analyze changing user interest over a period of time.

5. *How to replace cache*: This is another key problem. When cache is full, which content should be replaced first?. This could be done in First-In First-Out (FIFO) manner and it could be based on the freshness of the content.

6. *How to adjust different content size*: How to adjust variable content size in cache is also important, because different traffic types have different content size. Available node cache space and node positioning are the dependent parameters of content size.

7. *How to support mobility*: The mobile environment is significantly different from traditional client-server approach due to frequent disconnections and low bandwidth. These factors increase delays in case when a mobile sends a request for content and moves to a new location before receiving the content.

1.1.1 Possible Management Approaches

1. *Cache every where:* One primitive approach is to store everywhere on the path which is the default cache policy of CCN [10]. The main limitation of this approach is that it is not resource-efficient as every content have to be unnecessarily cached on each content router.

2. *Graph-based techniques:* Global topology-based techniques consider network related knowledge. Specifically taking into account the information regarding the physical location of a node while taking cache decisions. In this category, centrality-based caching algorithms are commonly employed, which include Degree Centrality (DC) [15], Betweeness Centrality (BC) [16] and Closeness Centrality (CC) [17]. Each of these algorithms is applied at each potential caching node on receiving every new content. The main limitations of these techniques are that they require complete knowledge about topology, i.e. the information about off the path neighborhood of a node, which is computationally expensive.

3. *Per-node feasibility based on local characteristics:* Storage space, willingness, power consumption, and cost per storage are considered as the local attributes of a node [18]. Although a node has high feasibility to store content, based on local characteristics but saving may lead to inefficient utilization with reference to the overall topology, e.g. distance from client.

4. *Content prioritization:* Saving every content is not wise decision so there is need to prioritize contents based on attributes such as popularity of content. However, content-based knowledge schemes consider content popularity independent of information related to network topology and local node characteristics. Hence, it is possible that the content is ultimately stored at every node or the focal nodes with the potential to serve a large number of clients are missing altogether.

Similarly, above techniques such as graph based techniques do not take advantage of content knowledge (i.e. content's popularity) for content caching. Therefore, it is possible that selected node might get overloaded with less useful contents over time.

1.2 Problem Statement

Intelligent cache placement involves many aspects, ranging from node's local characteristics to its positioning within the network. The importance or priority of each attribute may differ from scenario to scenario. When we use any parameter, it can appropriate in particular environments and may result in sub-optimal performance when the environment is changed. Thus, there is a need for such a solution in which no single attribute predominate and this solution is also dynamically adjustable and adaptable as per the environment. The decision about content placement is complex and a single attribute is not sufficient to take intelligent content placement decisions. For this reason, there is a need for multi-dimension analysis for taking intelligent content placement decisions.

We realize that in-network cache management is a multi-attribute optimization problem, which is accompanied with a dynamic weight adaptation mechanism based on error-correction learning to define relative importance of each attribute for an optimized combination. Secondly, in order to avoid single point of congestion it is also very important to distribute the load. Taking multiple attributes have the inherent advantage to distribute the load.

Moreover, cache management may become more challenging in CCN, especially, when it is accompanied with mobility related issues. In dynamic environments, mobile users experience long delay in communication because nodes may frequently get disconnected from the rest of the network and requester cannot access data from the data source. Therefore, ensuring data availability at the time of network partitioning becomes a demanding task. Another issue of mobile ad-hoc networks is about constraints on resources like bandwidth, cache resources, computational resources and limited battery power. It is of great significance to optimize, what contents to cache and where to cache by considering node mobility, content diversity and content popularity. Popular contents have to be replicated in multiple nodes and these contents should be relocated before network partitioning occurs.

1.3 Research Contribution

We propose a new multi-attribute caching scheme for CCN. The contributions in this thesis are summarized below:

- We propose a Multi-Attribute Caching Strategy (MACS) for CCN that determines suitable caching location along the content delivery paths while taking into account multi-dimensional analysis. MACS makes independent caching decisions at individual content routers. (Section- 3.2)
- To determine the relative importance of each attribute, a dynamic weight adaptation mechanism is proposed based on error correction learning [19], improving the scalability of MACS to add new parameters. (Section 3.4.5)

- Instead of only using router and network related characteristics, MACS functionality is further enhanced by considering a new dimension, that is content-related characteristic, while determining the score to store a content at in-network content router which shows significant improvement over previous work. (Section-3.4.3.2)

- We define a new method to dynamically update freshness of a content. Furthermore, we introduce a new cache replacement policy based on freshness value of a cached content. (Section-3.4.6)

- A Caching Strategy in CCN-MANET (CSCM) is proposed that considers the node's centrality, energy level and storage capability. This scheme dynamically adapts the caching decision of each content.(Section-5.2)

- In order to improve data availability and efficient utilization of network resources, this work design the popularity and freshness driven mobility adaptive cache relocation algorithm that relocates the replica if the old dominator moves to another region.(Section-5.2.3)

1.4 Research Methodology

Our research methodology consists of the following steps:

- Extensive review of existing caching schemes.

- Design detail illustrated with the help of flow diagram.

- Functionality description of the proposed work with the help of example scenarios.

- We use the open-source ndnSIM simulator [20] for evaluations.

- Performance is validated by running a number of simulations using different seeds and results are averaged.

1.5 Thesis Organization

The rest of the thesis is organized as follows: in Chapter 2, we present literature review of related caching strategies. Chapter 3 describes design details and elaborates description on our proposed caching scheme. Evaluation of MACS, using variety of scenarios, is explained in Chapter 4. Mobility-based caching strategy is evaluated in Chapter 5 and its evaluation and simulation results are also presented in Chapter 5. Conclusion and future work are presented in Chapter 6.

Chapter 2

Literature Review

2.1 Introduction

CCN architecture provides transparent and ubiquitous in-network caching. Aforementioned features result in speed up of the content distribution and improve network resource utilization. CCN cache introduces new features such as cache transparency, cache ubiquity and fine-granularity of cache contents as compared to the traditional cache systems [9]. In-network cache performance can be optimized in a number of dimensions that are described in Table 2.1 [9].

Table 2.1: CCN caching issues

Dimensions	**Issues**
Cache Size	How big the cache space should be to be assigned to a node to achieve a noticeable improvement in the performance?
Cache Sharing	How effectively share limited resources between different types of traffic?
Cache Space Sharing	Whether the sharing mechanism is fixed or dynamic?
Cache Content Decision Policy	Which objects will be placed in which cache nodes?
Timing of cache decision	Whether the cache decision is only made when a new object arrives or at the time when the cache is replaced?
Correlation between cache decisions	Whether cache decisions are made independently of each other or correlated?

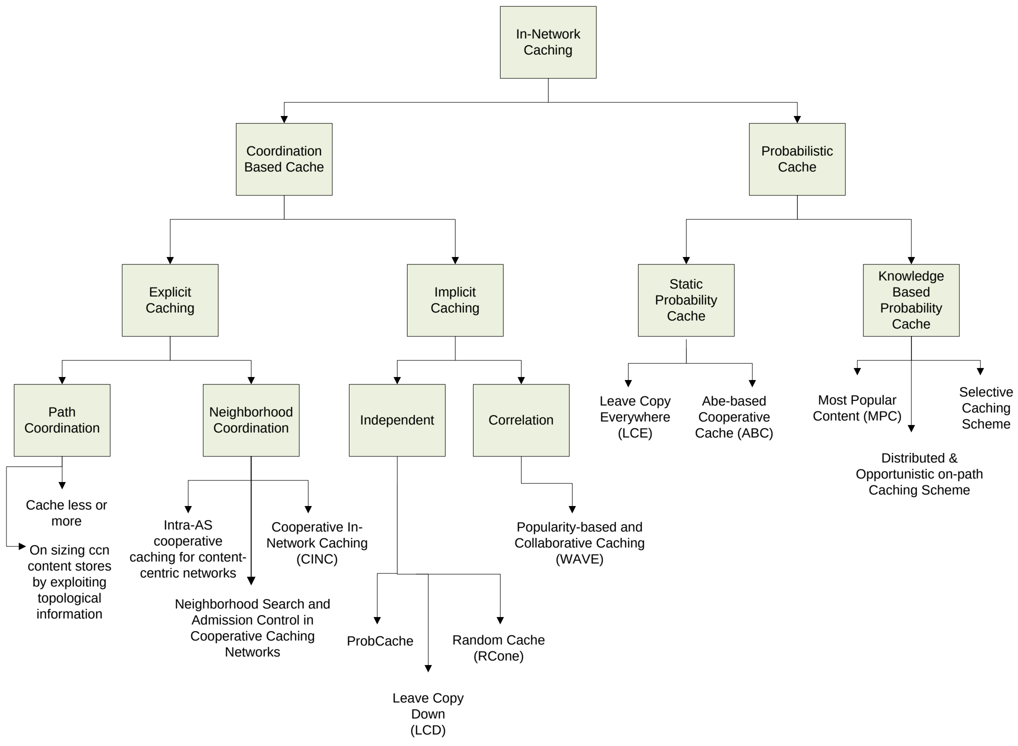

In-Network Caching
Coordination Based Cache
Probabilistic Cache
Explicit Caching
Implicit Caching
Static Probability Cache
Knowledge Based Probability Cache
Path Coordination
Neighborhood Coordination
Independent
Correlation
Leave Copy Everywhere (LCE)
Abe-based Cooperative Cache (ABC)
Most Popular Content (MPC)
Selective Caching Scheme
Distributed & Opportunistic on-path Caching Scheme
Cache less or more
On sizing ccn content stores by exploiting topological information
Intra-AS cooperative caching for content-centric networks
Cooperative In-Network Caching (CINC)
Neighborhood Search and Admission Control in Cooperative Caching Networks
Popularity-based and Collaborative Caching (WAVE)
ProbCache
Random Cache (RCone)
Leave Copy Down (LCD)

FIGURE 2.1: In-network Caching Taxonomy

The in-network caching strategies are proposed recently, that can be placed into many well defined categories. In-network caching performance is assessed in [21–28] using cascade or tree topologies. Moreover, D. Rossini et al. [29] and [30] describes the in-network caching performance for arbitrary networks. However, in this thesis we categorize the in-network cache schemes considering two perspectives which are based on the type of information employed in taking storage decisions, as shown in Figure. 2.1.

2.2 In-network Cache Schemes

2.2.1 Coordination-based Schemes

In coordination-based schemes, cache decisions are based on coordination among nodes. This type of caching schemes emphasize on node coordination to reduce cache redundancy and improve the cache diversity [31], [32]. Moreover, it also incurs control overhead over the network. Based on this overhead, we can further divide these schemes into explicit cache coordination schemes and implicit cache coordination schemes [9], which we explain in succeeding discussion.

2.2.1.1 Explicit Cache Coordination Schemes

Explicit cache coordination schemes, assume that every cache-router is equipped with prerequisite information about network topology, cache's state, user's access frequency. Therefore, these schemes are known to share large amount of information within the network. Table. 2.2 summarizes the proposed explicit-cache coordination schemes. Explicit cache coordination can be further categorized into Path Coordination and Neighborhood Coordination.

Path Coordination Schemes In Path Coordination Schemes [33], [34], nodes are selected to store a content based on coordination among all nodes present on delivery path from content provider.

In Coordinated Enroute Web Caching (CERWC) [34], every node on the path submit its candidature by embedding following information into request packet:

- State of cache at that node
- Access frequency of the requested object at that node

Hit node (i.e. node that contains the data) extracts all information from the request packet, and selects an appropriate candidate on the path. Every node on the delivery path examines the selected candidate node ID embedded in data packet, and saves the content if it matches with its own ID.

A large number of on-path caching techniques utilize global knowledge about network topology or information regarding physical location of a node while taking cache decisions [15], [17]. We collectively call them as graph based techniques.

These techniques use several metrics to measure the centrality of routers, including Degree Centrality (DC) [15], Betweenness Centrality (BC) [16] and Closeness Centrality (CC) [17]. Each of these algorithms is applied at each potential caching node on receiving new content.

In Degree-based Centrality caching mechanism [15] the number of links of each node is considered. Whereas Betweenness Centrality caching mechanism [15], [16] considers the number of occurrences of each node in many delivery paths. If a node has a high number of occurrences then this node is selected as a candidate node for caching contents. In Closeness Centrality-based mechanism, each node of the network calculates its shortest path distance to all other nodes and take the inverse sum of all shorter distance values. If this value is minimum then the node is more centralized. While, Degree-based Centrality algorithm selects those nodes which have highest centrality value, whereas closeness Centrality-based algorithm select those nodes which have lowest centrality value. Therefore, centrality-based algorithms may select only a precise set of nodes, which under utilized the network resources.

Moreover, a node's high traffic position in the network is ignored which may result in overloading that can cause high delays. Therefore, centrality-based algorithms are not taking advantage of content knowledge (i.e. content's popularity) for content caching. In other words, the efficiency of graph-based algorithm is topologically limited because it required topology manager that maintained centralized information.

Neighborhood Coordination Schemes Instead of taking information about all nodes within the path, in neighborhood coordination scheme, every node on the path only decides on information about its vicinity.

Along the same lines, the author in [35] proposed the Cooperative In-Network Caching (CINC) scheme based on a hash function, as shown in Figure. 2.2. This scheme does not cache all contents at a single router rather it stores only some contents of the stream. Every router is assigned a label which is a positive integer.

When a content comes at the cache router, then it first calculates its hash value to determine whether to store the content or not. It compares its calculated hash value with its label, if it is equal then the content is stored in this router otherwise the content is forwarded to the next router. Thus, it averts to store the same content at every router. When the request of a content comes at the router, then the hash value is calculated again to determine whether the content is stored on this router or not. If hash value is same, it means the requested content is available so the request for a content is not forwarded to the server, rather directly satisfied by this router. If the request is not directly satisfied by the router then this router forwards the request to its neighbors. Through this, it prevents the same object to be cached at all neighbor nodes. In [36], intra-AS

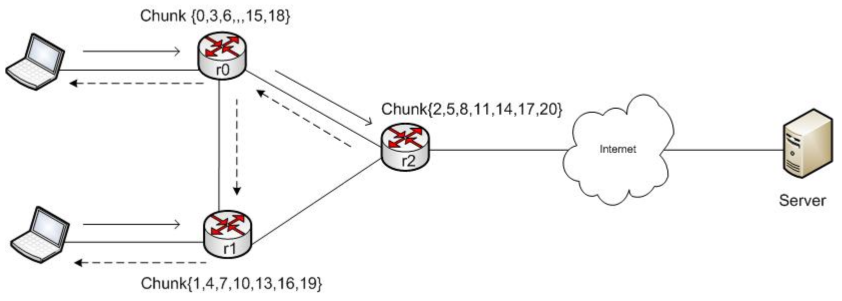


FIGURE 2.2: Coordinated In-network Caching [35]

cache cooperation scheme is proposed that solves the redundant caching problem by removing the duplicate content and improve the utilization of inadequate cache resources. This scheme allows one hop neighbors in an AS to cooperatively serve each others request. Periodically, neighboring nodes exchange the cache summary and eliminates duplicate items from its cache space. As a result, significant amount of cache space can be unconfined and it will be updated with fresh and popular contents.

Wong et al. [37] proposed a neighborhood search mechanism in order to increase the cache hit rate in coordinated caching networks. In this scheme, content requests are forwarded to the neighbors in the network path. Moreover, on-path routers maintain an available neighborhood routers mapping for efficient resource detection. Some benefits of this approach include traffic efficiency, opportunistic

Table 2.2: Comparison between explicit coordination-based schemes

Schemes	Coordination Manner	Parameters	Cache Redundancy	Knowledge basis
CERWC[34]	Path explicit	Cache gaining	Low	Content & node-based
DC[15]	Path explicit	Degree centrality	Depends on centrality	Topological centrality
BC[16][39]	Path explicit	Betweenness centrality	Depends on betweenness	Topological
CC[17][39]	Path explicit	Closeness centrality	Depends on closeness	Topological
CINC[35]	Neighborhood explicit	Chunk identifier	Low	Content based
Intra-AS[36]	Neighborhood explicit	No	Low	Content based

multi-source content retrieval and gradual deployment. Nevertheless, this scheme cache the same contents on every router of that path, as a result, it does not improve the hit rate. Additionally, in this neighborhood search mechanism, the content router also saves the index values of other content routers that have the content. Thus, these mechanisms entail additional cache space to store this information. In [38], general idea of a neighborhood searching mechanism is improved by introducing a new admission policy that increases the usage of storage space. This cooperative cache scheme also proposes a new neighbor search algorithm that stores neighborhood information using Bloom filters. Path and neighborhood coordination schemes reduce cache redundancy to a great extent, but introduce high computational complexity. Moreover, they require extensive information exchange which may overburden the network with control traffic.

2.2.1.2 Implicit Cache Coordination Schemes

Implicit cache schemes do not assume prerequisite information, about the cache network topology, to be available at every node. Instead they rely on only small

TABLE 2.3: Comparison between different implicit coordination-based scheme

Schemes	Coordination Manner	Parameters	Cache Redundancy	Probability Value	Cache Decision
LCE[43]	No coordination	No	High	Static value	Independent
LCD[44]	Implicit	Popularity	Medium	Static value	Independent
RCOne[40]	Implicit	Random decision	Low	Random value	Independent
Prob[18]	Implicit	Popularity	Depends on p	Static value	Independent
Opportunistic On-Path Caching for NDN[45]	Implicit	Popularity & distance	Low	Probability value	Independent
ProbCache[46]	Implicit	Distance between cached node & requesting node	Medium	Probability value	Independent
WAVE[47]	Implicit	File-level object popularity	Low	Static value	Correlated

information to be exchanged among cache-routers to make the final decision on where to cache the content [40]. A number of implicit cache coordination schemes have been proposed [40], [41], [18], [42]. Table. 2.3 summarizes the implicit-cache coordination schemes. In following, we present a review of existing well known implicit cache coordination schemes. In LCD caching scheme [44], a copy of the requested content is always cached at the next downstream node, as shown in Figure. 2.3. In this way, a content that receives a large number of requests is gradually pulled down within the network. Apparently, this schemes favors popular contents to be stored within the network, while less useful contents are only saved at edge routers. Moreover popularity is decided in an indirect manner based on increasing number of requests, which makes it computationally feasible. In Random Copy One (ROne) cache scheme, requested content is stored randomly

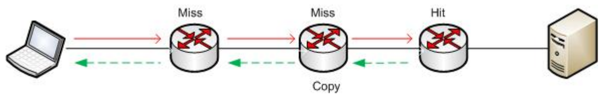


FIGURE 2.3: Leave Copy Down [9]

at one node along the downloading path. This caching strategy has the equal probability of caching contents at any node. Eum et al. [40] is an example of unique cache, which is called Random Copy One (RCOne). This RCOne approach equates the probability of caching at each node with $p = 1/n$ (where n=number of intermediate nodes), shown in Figure. 2.4. RCOne provides the same probability for both popular and unpopular contents, so it does not considers the content's popularity.

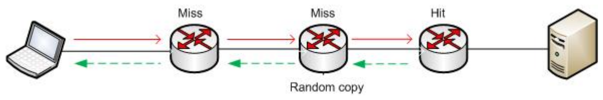


FIGURE 2.4: Randomly Copy One

In literature, Probabilistic Cache (ProbCache) scheme [18] is also proposed, in which requested content is cached at every node with probability p. Probability p is calculated at each node lying on the delivery path, according to the Equation 2.1. In this scheme, cache probability of every node varies because cache probability is inversely proportional to the distance from the requesting node. ProbCache is based on TimesIn factor and the CacheWeight factor.

$$ProbCache = \underbrace{\frac{\sum_{i=1}^{x-(y-1)} N_i}{R_c}}_{\text{TimeIn}} \times \underbrace{\frac{x}{y}}_{\text{CacheWeight}} \tag{2.1}$$

According to the ProbCache scheme, TimeIn factor is considered as the number of replicas of the content that a given delivery path has to cache. Where N_i is the cache size along the path and R_c is the number of the content items that processed on each node. Whereas, CacheWeight factor proceeds as a balancing factor to this unfairness. CacheWeight is calculated as $\frac{x}{y}$ where x and y correspond to the number of hops traveled from the requester to the content source and to the number of hops traveled from the source towards to the requester, respectively. However, the main purpose of ProbCache is to provide fairness to the available capacity of the delivery path.

Consequently, if the distance between the node and requesting node is small then the probability of content cache is high otherwise it has a low cache probability. The advantage of this technique is that it caches the content at the network edge with high probability which reduces the cache redundancy. Therefore, it reduce the cache redundancy and network traffic redundancy, but this technique introduces high computational cost.

Xiaoyan, HU et al. [48] proposed a lightweight distributed and opportunistic on-path caching scheme. In this caching scheme, each on-path router calculates the probability of caching a content, which is based on content's popularity and the distance from the content server to the content client. In such a way, most popular contents cached near clients routers that reduced the user's requests traversing. This on-path opportunistic caching scheme does not require any explicit statistics, which are exchanged among routers. Therefore, it improves the effectiveness of on-path caching and reduces the cache redundancy. [46] is another example of probabilistic on-path caching, where on-path available storage space is allocated according to the characteristic of traffic flow.

Kideok et al. [47] proposed a lightweight content caching scheme, called WAVE: Popularity-based and Collaborative In-network caching for Content-Oriented Networks. In which, content is cached only on the basis of content popularity information instead of network topology information. In this popularity based caching approach, an upstream routing node explicitly prescribes the number of chunks to be cached at its downstream node by marking each chunk. Consequently, as the number of requests increases about a specific content, copy of the corresponding content exponentially increases. In WAVE, when file is accessed first time, then, according to the algorithm, the first chunk of the file is marked to be cached. When the same file is accessed second time, then next two chunks will be marked. In a similar way, file chunks to be marked increase exponentially as file access frequency increases. On the other hand, low file access count will consume very few network resources. Along the same line, Abani et al. [49] proposed popularity-based caching scheme that progressively cache file chunks as the file gets more popular.

In the aforementioned implicit caching techniques, a content is cached only at a single node along the delivery path or a popular content is eventually, unnecessarily stored on most of the nodes within the network. Hence, these techniques can lead towards high content redundancy or no caching at all.

2.2.2 Probabilistic Caching

In order to avoid coordination overhead while still making intelligent decisions regarding in-network caching and to improve end-user quality of experience– probabilistic caching is introduced. In probabilistic cache schemes, every node independently calculates its feasibility to store a content. To cache the content, decisions are based on some probability value p. The probability p can be derived as a static random value or may be calculated on the basis of some locally available knowledge [50], [47], [51].

Probabilistic caching schemes can be further classified as random static probability caching scheme and knowledge based probability cache schemes. We first describe random static probability caching scheme.

2.2.2.1 Static Probability Cache

The static probabilities caching scheme does not consider any knowledge about the content, just probability value p is statically set according to the standardized value (i.e. $p = 1, 1/2, 1/3, etc.$) [50]. In this scheme, each node takes the caching decision based on static probability value p and there is no coordination between nodes.

In Leave Copy Everywhere (LCE) caching scheme, as data packet traverses through the network, each content router on the downloading path can potentially cache the content [10], which is default CCN caching policy, shown in Figure. 2.5. This avoids the need to send back every new request for the same content to the server, which helps in reducing the load over servers. Although it increases content availability and quality-of-experience (QoE) to end user by reducing delays, the approach is not resource-efficient.

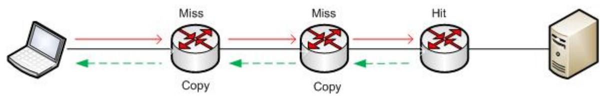


FIGURE 2.5: Leave Copy Everywhere [10]

Z. Ming et al. proposed an age-based cooperative cache scheme [52]. This cache scheme is another type of LCE. However, it additionally attach an age to every content, after which the content is automatically removed to free node cache. This age is automatically calculated at every node between the subscriber and provider. In this scheme, most popular contents are disseminated on the edge router while the less popular contents are on the intermediate routes and so on. For the age based cooperate cache, age is calculated on the basis of two parameters such as distance and the popularity of the contents. If content has lager popularity and have the longest distance from the server then the age is higher otherwise age is shorter. Through this, most popular contents on edge router exist longer than less popular contents on intermediate routers. Intermediate router removes the contents when the age of the content is expired or cache is full. Figure. 2.6 shows the work-flow of age-based cooperative caching. Clients and server are connected by router $R1$ and $R2$. The server has two contents: $c1$ with high popularity and $c2$ with low popularity. Client requests the content $c1$ and server serves the $c1$ and then both routers $R1$, $R2$ cached the content (Figure. 2.6(b)). According to the age-based cooperative cache scheme, router $R2$ assigns the less age as compared to router $R1$ because, age is calculated on the basis of two parameters such as distance and the popularity of the contents. After that, the content $c1$ expire at router $R2$ but still exist at the router $R1$ (Figure. 2.6 (c)). At the same time, if client request the content $c2$ then content $c2$ is automatically cached at router $R2$ (Figure. 2.6 (d)). The governing principle behind this scheme is to spread popular contents to the network edge and remove the unnecessary content replication on intermediate routers. Therefore, this scheme reduces the network delay and publisher load. On the other hand, there is no signaling messages between routers for the age of a content and it introduces less computational overhead.

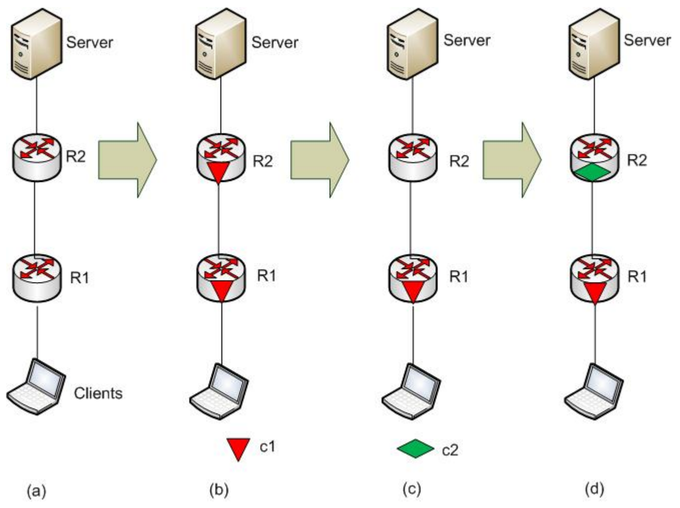


FIGURE 2.6: Age-based Cooperative Cache Workflow [53]

2.2.2.2 Knowledge Based Probability Cache

In knowledge based probability cache scheme, cache decisions are based on some probability value p, which can be calculated on the basis of some knowledge about either the content or node position within network topology in terms of its degree of connectivity. Probability cache is shown in Figure. 2.7. A number of knowledge-

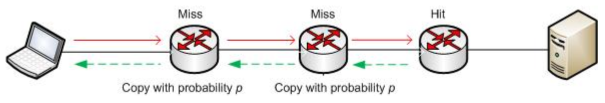


FIGURE 2.7: Copy with Probability [9]

based probabilistic caching strategies are proposed in the literature that attempt to reduce redundancy of cached copies [18, 46–48, 54–58]. Hence, knowledge-based schemes intend to make careful decisions for intelligent utilization of constrained network resource. These schemes can further be divided into following two categories based on kind of attributes employed in calculating the probability to store the content such as: 1) Content-Based knowledge and 2) Topology-Based knowledge. Content-Based knowledge technique considers information that is relevant to content etc. (i.e. content's popularity or freshness of the content etc.). C. Bernardini et al. proposed a caching strategy called Most Popular Content (MPC) [54], where only popular contents are stored which is shown in Figure. 2.8.

Popularity of the content is measured through frequency of interest messages. For the popularity measurement, every node counts incoming request messages for a specific content and maintain a popularity table. When content requests reach content popularity threshold, then it means it is popular and the nodes will cache the popular content and send a suggestion message to its neighbors to cache this content. This scheme reduces cache redundancy so as to save the resources of the caches.

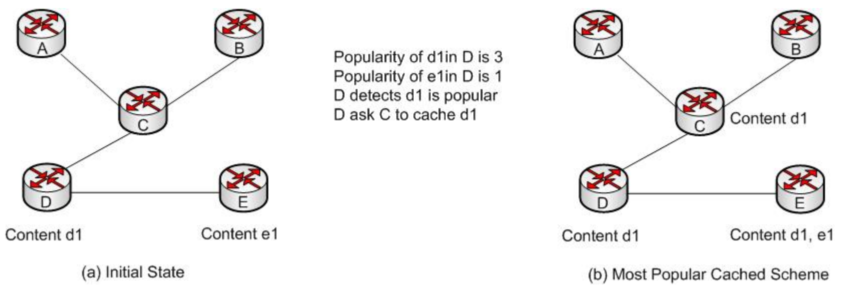


FIGURE 2.8: Most Popular Cached Scheme [54]

Guoyin et al. proposed an Optimal Cache Placement strategy [59] based on Content Popularity (OCPCP). OCPCP improves the cache hit and reduces redundancy of the cache contents. Moreover, M. Bilal et al. proposed Conditional Leave Cope Everywhere (CLCE) and Least Frequent Recently Used (LFRU) [39] that is cache replication and eviction schemes for content-centric networks. This scheme improves the utilization of cache space and reduces the redundant caching. [46] is another example of probabilistic on-path caching, where on-path available storage space is allocated according to the characteristic of traffic flow. Another study explores the potentials and limits of a probabilistic caching scheme in content-centric networks [60].

Takahiro et al. proposed a selective caching scheme [61] that cache only those contents which are frequently requested. This work utilizes the accumulated content request history on the content router to determine the chunk likely to be used next. More particularly, each content router calculates the weighted average using exponentially weighted moving average (EWMA) mechanism to determine the updated content popularity. On every router, the content request history list

Table 2.4: Comparison between different probabilistic caching schemes

Schemes	Coordination Manner	Parameter	Cache Redundancy	Probability Value	Knowledge Basis	Parameter
LCE[43]	No coordination	No	High	Static value	No	No
LCD[44]	Implicit	Popularity	Medium	Static value	Content	Popularity
Age-based Cooperative Caching in CCN[53]	Implicit	No	Medium	Static value	Content & distance	Popularity
MPC[54]	Implicit	Popularity	Depends on p	Calculated value p	Content	Popularity
OCPCP[59]	Implicit	Popularity	Depends on p	Calculated value p	Conetnt	Popularity
LFRU[39]	Implicit	Popularity	Depends on p	Calculated value p	Content	Popularity
Selective Caching Scheme[61]	Implicit	Request History	Depends on p	Calculate value p	Conetnt	Request History
LeafPopDown[57]	Implicit	Popularity	Depends on p	Calculated value p	Content	Popularity
EEACC[58]	Implicit	Content Priority & Energy Efficiency	Depends on p	Calculated value p	Content Content	Popularity & Node centrality

is maintained by using the following Equation. 2.2.

$$E_c = \alpha R_c + (1 - \alpha) E_c \tag{2.2}$$

Where E_c is a request history list and R_c is the number of receiving requests for the chunk c. The candidate router that receives the content, first checks the request history list, then decides either to store the content or not. If the content is exists in the candidate list, then it stores the content otherwise it just forwards the received content.

Table. 2.4 shows the comparison of different probabilistic in-network caching schemes indicated so far from several aspects such as coordination manners are implicit or explicit, cache decision basis, the degree of resulting cache redundancy, probability value of p and content & topological knowledge value.

We can see that aforementioned schemes focus only on content related attributes or distance from content provider or the client on the downloading path. However, a router's own characteristics in terms of available free space or degree of attachment to other nodes within the network are ignored which may result in degrading performance in different scenarios. Hence, we suggest the need to calculate feasibility of a router to store a content, based on multiple parameters. The proposed solution is equipped with the ability to dynamically learn network and content related characteristics to better utilize available resources which makes it suitable for diverse environments.

2.3 Mobility and Cache Management in CCN

In this section, we present a background and related work of Information-Centric caching in Wireless ad-hoc network. Figure. 2.9 illustrates the caching framework based on Mobile ad-hoc networks, in which requested data is retrieved in a multiple-hop fashion.

2.3.1 Caching in Mobile Ad-hoc Networks

Dynamic topology, characterized by short-lived contacts, is one of the crucial components of wireless networks. CCN for wireless ad-hoc networks is introduced to improve the network performance and to reduce the impact of dynamic topology. Due to the mobility of mobile nodes, partitions in the network happen due to which the requester may not be able to access data item from the data source. In this case caching can help to improve the accessibility and availability of data items. The default CCN architecture does not make specific assumptions on cache decision, however, the related literature has typically considered that all nodes may cache all new contents that increase the data accessibility, decrease the retrieval latency and low the redundant requests. However, the resource inefficiency of the above mentioned approach makes it unstable to be used in MANET. In MANET, data caching strategies is extensively used to cope up with the resource constraints and plays a significant role in improving the accessibility of data contents. There is a threat in wireless environment that indiscriminate caching may waste network bandwidth and device energy as the result of multiple transmissions of cached contents from many nodes. However, the issue of efficient caching in wireless networks is closely related to the content cache placement and distribution in CCN [62]. The main factors that affects the performance of mobile content cache placement policies are as caching redundancy, policy complexity, content popularity, content granularity, mobility of caching networks and whether caching nodes cooperate with each other or not etc., due to the mobility of users and constrained transmission bandwidth in MANET [63].

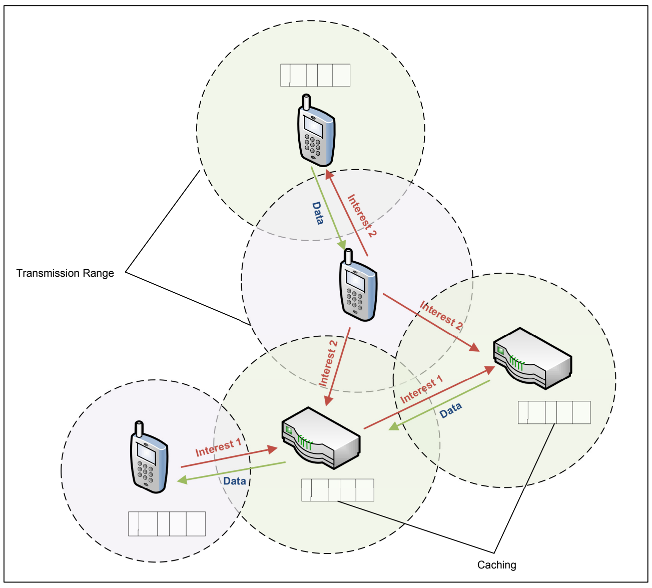


FIGURE 2.9: Caching Framework based on Wireless ad hoc networks

Number of cache placement policies are proposed in CCN-MANET that can be placed into many well-defined categories, as shown in Figure. 2.10, such as, 1) Genral Caching, 2) Cooperative Caching, and 3) Mobility-Aware Caching.

2.3.1.1 General Caching Techniques

General caching techniques are distributed to place the cache, there is no coordination among nodes and an independent caching decision is based on some parameters or criteria. These strategies are also further classified and this classification is based on the amount of knowledge they require to compute the caching decision, that are 1) Simple Caching, 2) Location-aware Caching, and 3) Probabilistic Caching.

Yuguang et al. examined the impact of a caching strategy on the named data delivery in MANET [62]. In this caching strategy, some nodes are selected (i.e.

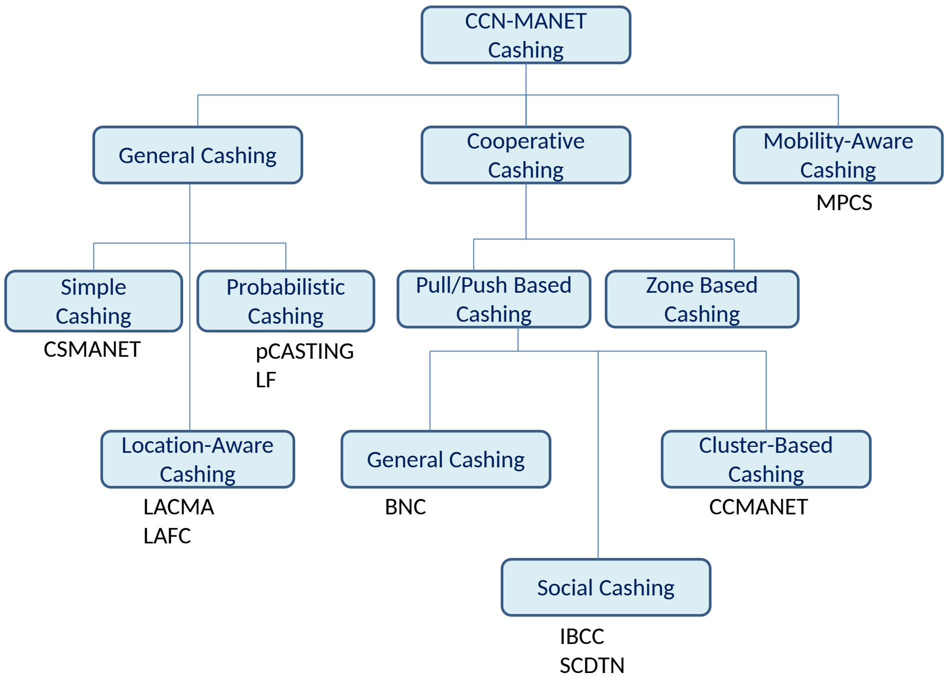


FIGURE 2.10: CCN-MANET Caching Taxonomy

based on data interval parameter that is use in *Data* packet) to cache the content along the downloading path that reduces the overhead and the data redundancy in NDNs. Therefore, this strategy is easy and low cost.

However, location-aware caching schemes take into account the location information about consumer and provider nodes while sending *Interest* and *Data* packets. In [64], strategic content placement problem is highlighted in a dynamic MANET. The proposed on-demand location-aware forwarding and caching scheme considers the distance and remaining energy for caching decision. LACMA [65] is another example of location-aware caching approach, where bind data to geographic location. Location-Aided Content Management Architecture (LACMA) based on the location information that tries to keep a content copy within a specified geographic boundary by proactively replicating the content as necessary.

The author proposed a novel probabilistic CAching STrategy (pCASTING) in a wireless NDN-IoT network [66]. This scheme relies on the freshness of data and on potentially constrained capabilities of devices. It dynamically adapts the caching probability at each node. Author calculates the caching probability by considering content parameters and device parameters such as freshness of data, battery energy level and the cache occupancy of the node. In order to select where to cache, in traditional caching techniques, especially, strategic content placement approaches are considered to favor less space utilization. Furthermore, the author proposed less space still faster (LF) caching strategy [67], where caching decision is based on the popularity of data content, available storage space in the node and the distance of the node from the data source. Therefore, in LF just some nodes are selected for cache popular data along the forwarding path, cache node selection criteria is based on computed caching value Vi using the following formula.

$$Vi = \gamma(\alpha h + \beta B^p) \tag{2.3}$$

Where α is its weight coefficient, h is the hop count of the interest packet, β is its weight coefficient, p is the popularity and B is the base (2 is used) argument of the popularity.

2.3.1.2 Cooperative Caching

Cooperative caching for mobile ad-hoc networks has also been studied widely in recent years [68], [69], [70] which allows sharing and coordination among multiple caching nodes. There are two cooperative caching techniques in MANET such as Push and Pull based scheme, and Zone based cooperative caching scheme [71].

Push and Pull Based Caching Scheme: In the push-based caching scheme, a node actively advertises its newly cached data information to its neighboring nodes and these neighboring nodes will record the caching information upon receiving an advertisement. These nodes use this information to redirect the upcoming number of requests for the same content. So, the scheme enhances the usefulness of the cached content. But in case, if there are no requests arises for the cached content from its neighboring nodes then the advertisement of cached content is useless. However, the cost of this scheme is communication overhead due to advertisements. Additionally, the cached information may become obsolete due to node mobility or cache replacement. Pull-based scheme may overcome this problem. In the pull-based caching scheme, a node will broadcast a content request to its neighboring nodes when the local cache miss occurs and it wants to access the content. In response, a node that has the requested content in its cache will send the copy of the data to the requesting node. So, the pull-base scheme allows the nodes to utilize the latest cache contents.

Pull-based caching scheme can be further categorized into General Caching, Cluster Based Caching and Social Based Caching.

General Caching:

Broadcast-based Neighborhood Cooperative caching (BNC) strategy was proposed for content-centric ad-hoc networks [69]. In BNC, each node used topology potential method to select a Cache Node (CN) among its adjacent nodes, which indicates the importance of a node in the network. BNC technique achieves neighborhood cooperative caching by using the advantages of the broadcast feature of wireless channel. The results showed that BNC can be a better utilization of network cache space.

A hierarchical cooperative caching scheme is proposed for mobile nodes [72], in which buffer space of mobile node is divided into three key components: self, friends, and strangers. The mobile node caches the most frequently access data items in its self-caching part. The mobile nodes with higher contact frequency are considered as friends to the cache node. In friend's part, the friend's most frequently accessed data item is cached. Moreover, mobile node randomly caches the remaining data items in the stranger's part.

Cluster Based Caching:

Cooperative caching schemes consider content popularity to keep more popular contents in caches. A novel cooperative caching scheme CCMANET was proposed [73]. This scheme is based on generalized dominating set and local content popularity where generalized dominating set is used to construct a virtual backbone in CCMANET and divide the arbitrary topology into a two-level hierarchy architecture. Another scheme that is collaborative cache placement scheme was designed between these two-level hierarchies. Author also defined the method of local content popularity computing and cache placement scheme that is based on local content popularity. However, this paper mainly focused on how to improve the performance of caching by selecting the appropriate nodes and it introduces high complexity in a huge network.

Social-Aware Caching:

To overcome the constraints of content-centric dissemination, research community proposed some new solutions that are based on social characteristics of user's behaviors.

Researchers have proposed social caches that share information within a social environment [74]. A social cache is a logical collective view of individual node that caches contents that are useful to the members of its social community. Social community spends significant time together, meet more frequently and to be willing to cooperate with each other. By analysis such social relations or mobility patterns, nodes can be identified that are more suitable for caching content. The proposed protocols of social-aware dissemination are based on all these aspects and therefore enhance the performance of content dissemination.

Recent studies about socially aware data distribution fall into two categories [75] 1) based on solicitation and cache, from node's point of view 2) based on forwarding, from the content perspective. In solicitation and cache based category, a node can cache uninterested data items for other nodes to improve the content distribution efficiency. However, the cache size of the mobile node is limited, so it is impractical to cache all encountered contents. Frequent buffer replacement will induce significant power consumption and decrease the efficiency of the network. Hence, according to local and/or global environments, soliciting and caching the appropriate set of contents for future dissemination is an effective way to reduce energy consumption and to improve the efficiency of data distribution. The second one is based on forwarding, from the content perspective. In social communities, people share common interests for specific content and their social activities facilitate information sharing and communications among them. For successful data dissemination, it is mandatory to search the proper forwarder that can disseminate the content to the destination community as fast as possible.

The author proposed and evaluate ContentPlace scheme [76], that exploits dynamically learn information about user's social relationships to decide where to place data objects. This scheme considers five policies to evaluate the social weight such as most frequently visited (MFV), most likely next (MLN), future (F), present (P), and uniform social (US). ContentPlace does not need any overlay infrastructure but it utilize the same community detection mechanisms as in [77]. Basically, nodes independently decide which data objects should be cached, based on assigned each data object a utility value. This utility values is defined by the size of the content, the probability that the content is available and the probability that the node can get access to that content. Therefore, nodes exchange their summery vector when they come into contact. Summery vector contains the lists of items in one anther's cache and then the items to be stored are selected among these listed items. In ContentPlace, the selection of contents to be cached is based on the maximum value of local utility of its cache by solving the multi-constrained 0-1 knapsack problem. In the same spirit, the author presents a routing protocol for publish-subscribe that relies on prediction of node popularity for taking caching decisions [73]. Those nodes are preferred as content carriers that have higher importance in the community. Social aware dissemination techniques do not rely on underlying contact or content popularity rather they directly get this information through online estimation process (as in ContentPlace). This implies that social-aware content dissemination strategies are expected to be resilient to mobility and content popularity changes. A major drawback of this type of dissemination techniques is the overhead that is created by the process of collecting and managing the information.

Recently research community focuses on the social aspect of caching in CCN [78],

[79], [80] and indicates that these social aspects critically affect on content access patterns. The users in the same social community might be interested in the same content. Therefore, the contents proactively pushed to the cache of user's immediate neighbors. These immediate neighbors are may be 1-hop neighbors or a specific ratio of common content items with the user [79]. Bernardini et al. proposed a social based caching strategy for CCN [78]. Socially-Aware Caching Strategy (SACS) gives priority to content issued by influential users and proactively caching the content they produce. Influential users are evaluated by using Eigenvector [81] and PageRank[82] centrality measures. This caching strategy understands how users interact and react accordingly.

In [80], proposed a caching strategy that is called a Socially-aware NodeRank-based Caching Strategy (SNCS). SNCS designed a ranking algorithm for nodes and Key nodes which are selected according to their ranking scores in CCN. The ranking score is derived from user's social behaviors with physical information and choose the highest ranking node to cache copies of each sociAS along the request path. Hence, key nodes will proactively disseminate popular contents to consumers and alleviate server's pressure.

The social-aware techniques for CCN are defined for static networks that are not suitable for MANET or opportunistic scenarios [78], [80]. Moualla et al. presented a push-based proactive content replication strategy which takes advantage of the caching and mobile content-centric ad-hoc networking capabilities of user devices [79]. In this approach, distributed mechanism is designed to estimate the social distance between users and to find the popularity of content items. Bloom Filters are used as summaries of user caches and then define the socially aware per-fetching scheme that exploits information of user cache contents, instead of external information such as data from user past encounters, activities in existing social networks etc., here users proactively push popular content to the caches of their peers.

Moreover, another study Socially Aware Vehicular Information-Centric Networking model (SAVING) capturing the research community interest towards an application of combining socially aware content distribution scheme with the information-centric networking paradigm [83]. Mobile nodes such as vehicles often move to places where they can establish communications and SAVING integrated three classes such as Computing, Caching and Communications (3Cs) to enhance data dissemination in a content-centric mobile network. In computing class, a mobile node exploits social information to compute its eligibility to be selected as a potential candidate in order to cache content data. To relay and retrieve cached content in the network, SAVING incorporates the ranking system that comprising three novel centrality schemes such as InfoRank [84], CarRank [85], and Grank [86]. InfoRank system classify different cached content and this classification is based on its availability popularity and timeliness with respect to the user interest. After that, the vehicle independently computes its relative importance in the network using CarRank and Grank system. The caching class involves different cooperative caching schemes where identified important node efficiently caches the content based on the content popularity and availability in the network. Communication class incorporates a socially-aware content distribution protocols, where cached content is retrieved from the important information node in the network.

Zone Based Cooperative Caching Scheme One hop neighbors of a client node are included in a cooperative zone cache [87], so in this case the cost of communication is low in terms of message exchange and energy consumption. In this scheme, client node shared its cache data with its one hop neighboring nodes. Cache discovery is a problem in cooperative zone cache, when a node initiated a request for a content, first, it look-up content in local cache if it is miss then the node check the data in its neighboring node's cache within its home zone. When neighboring node receives the request and cache hit occur, it will send the copy of the requested cached data to the requester. If the zone cache miss occur then the request is forwarded to the neighbor node along the routing path. If the data is not found in the neighboring zone, then the request is forwarded to the data server and it sends back the requested data.

Le et al. proposed a social-based cooperative caching scheme [88]. The main idea of this scheme is to dynamically cache popular content data at cluster head nodes. These are popular nodes along the common request forwarding paths that have the highest social level in the network. In this scheme, neighbors of downstream nodes may also be involved for caching due to limited caching space of buffers in the mobile nodes. If the cache of downstream nodes is full then some existing data is moved to its neighboring nodes to accommodate new data. Furthermore, author describes the dynamic cache replacement policy based on the content popularity that considers both the frequency and freshness of data access.

Another cooperative caching cached data on Network Central Locations (NCLs) nodes [89]. The selection of NCL is based on data type. The coordination between caching node aims to optimize the trade off between data accessibility and caching overhead. The proposed strategy tries to overcome the problem of data transmission rate and the disruptions.

The cooperative caching is helpful to solve the use of network bandwidth and query delay to retrieve the data from the server and reduced the energy consumption.

2.3.1.3 Mobility-Aware Caching

In literature, some studies consider mobility issues which have been studied in the area of CCN-MANET [90].

Wei et al. proposed a Mobility and Popularity-based Caching Strategy (MPCS) for CCN [90]. MPCS aims to increase the cache hit rate when people are moving from one WiFi hotspot to another. In this scheme, there are two caching strategies that are defined as MOC(Mobility-Oblivious Caching) and MAC(Mobility-Aware Caching) by using the advantage of user mobility pattern and content popularity. In MOC, popular contents are stored locally at any site to obtain an optimized cache hit rate, whereas, in MAC, cached contents are selected from the popularity rank list by considering both the popularity rank list and the site transition matrix.

In [91], proposed a Popularity-driven Caching and Mobility adaptive query Searching (PCMS) scheme to overcome the constraints such as reducing the data access time, energy cost, and overhead. There are three components in PCMS such as Cache Node selection for popular data and data size driven replacement algorithm and mobility adaptive searching for cache data. In Cache Node selection algorithm, a node is selected as a Cache Node that forwards several requests to different destinations for the same data. This technique extracts the data popularity from the routing information. PCMS uses the data popularity and data size information while taking cache decisions and removes the unpopular data from the cache for effective usage of the cache memory of the node. Whereas in mobility adaptive searching for cache data, a mobile node receives a request from other mobile node and search an item in its Query Directory (QD) and redirects the request to the concerned Caching Node. Query Directory (QD) information is updated by adhering to the mobility prediction scheme.

In the following table we differentiate among different schemes based on features such as performance metrics, content granularity, replacement, advantages and cooperation among cache-enabled network elements. Table. 2.5 illustrates the comparison of different cache placement policies.

None of the above schemes take into account the availability of data at the point of network partitioning that is an important factor in MANET. Particularly, when mobile nodes move and often network partitioning occurs and data in two separate networks to become unreachable to each other. On the other hand, constraints on resources like bandwidth, cache, computational resources and limited battery power in mobile ad-hoc networks, it is a great significance to optimize what contents to cache and where to cache considering node mobility, content diversity and content popularity in order to achieve ideal performances.

To address these issue, in advance popular data have to be relocated in multiple nodes before network partitioning occurs. However, the proposed strategy is

Table 2.5: The comparison of different cache placement policies

Policy Type	**Parameters**	**Performance Metric**	**Cooperative**	**Replacement**	**Content Granularity**	**Problem**	**Contribution**
A Caching Strategy in Mobile Ad Hoc Named Data Network	Data Interval	Average Query Delay, Average Path length, Cache Hit Ratio, Overhead	No	LFU	Content Level	CCN used in MANETs, will render the network with too many duplicate contents	The redundancy of contents in MANET is reduce
Location-Aware Forwarding and Caching in CCN-Based Mobile Ad Hoc Networks	Distance Energy	Content retrieval time, interest retransmission	No	LRU	Content Level	Due to the broadcast nature of the wireless channel, various issues such as packet flooding, data redundancy, packet collisions, and retransmissions etc.	Reduce data redundancy & delay, avoid packet flooding
Location-Aided Content Management Architecture(LACMA)	Location information & Content popularity	Reducing number of hop counts & query traffic	No	Not define	Content level	Content placement optimization problem	Define location based cache placement strategy
pCASTING	Freshness of data, energy level and storage capacity	Cache hit ratio, Consumers' received data pkts, Data retrieval delay[s]	No	LRU	Content Level	IoT devices can be resource-constrained, with limitations in terms of energy, storage and processing capabilities.	Design a caching strategy specifically tailored to wireless multi-hop information-centric loT systems, Reduce the energy consumption and delays
LF: A Caching Strategy for Named Data Mobile Ad-Hoc Networks	Distance, available space, Popularity	Response time, network traffic, and cache hit ratio	No	LRU	Content Level	Storage space of nodes in a MANET is limited, the on-path caching strategy of NDN must be modified	Define less space still faster (LF) caching strategy, Reduce overhead and improve caching efficiency
A Novel Cooperative Caching Scheme for Content Centric Mobile Ad Hoc Networks (CCMANET)	Content popularity	Path stretch and server load, and hit ratio	Yes	Cache replacement scheme	Chunk Level	Caching scheme in CCMANET has not been well explored	Reduces both path stretch value and server load, & improves the hit ratio
Broadcast based neighborhood cooperative caching (BNC)	Topology potential	Content retrieval latency, hit ratio	Yes	LRU	To improve the caching gain in CCN for MANET	Content Level	Better utilization of network cache space
Interest-based cooperative caching in multi-hop wireless networks	Social-distance user interest	Cache hit probability and access delays	Yes	LRU	Content Level	Content placement problem	Improve caching efficiency

based on relocation model in CCN-MANET that relies on node's current status and node's centrality in the network. Moreover, we consider content placement optimization problem in wireless environment.

Chapter 3

Multi-Attribute Caching Strategy for Contnet-Centric Networking-Design Details

3.1 Introduction

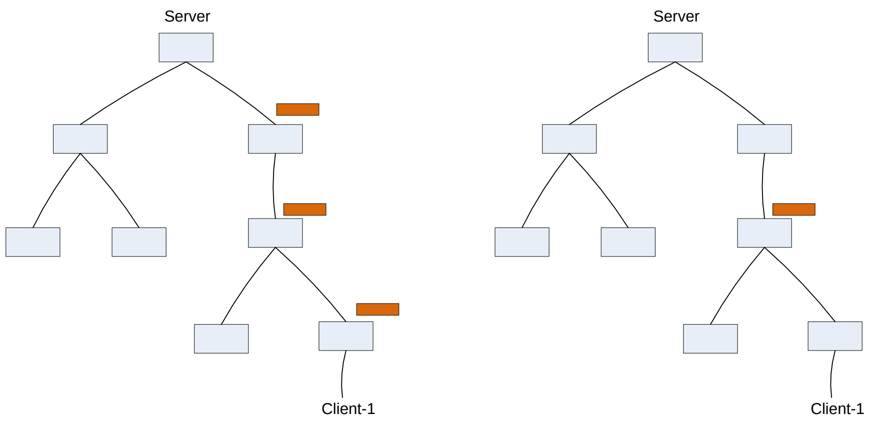
CCN utilizes the concept of in-network caching in order to enhance the availability of content in the network. As data packet traverses through the network, each content router on the downloading path can potentially cache the content, as per the default cache policy of the architecture. While the idea helps in increasing content availability and quality-of-experience (QoE) to end user by reducing delay, and reduces load over servers, the approach is not resource-efficient as it introduces high cache redundancy, i.e. cached the content unnecessarily at multiple neighboring nodes. Hence, there is a need to devise efficient caching mechanisms that allows maximum availability of content while consuming minimum possible resources.

In order to address the problem of efficient content utilization, some approaches emphasize on node coordination to reduce this cache redundancy and improve cache diversity [14], [21], [22], [23], [50], [25]. There have been many studies targeting implicit in-network caching [52], [54] and explicit in-network caching schemes [34], [35]. In explicit cache coordination schemes, every cache-router requires pre-information about cache's state, user's access frequency and exchange a lot of information. Therefore, explicit caching schemes reduce cache redundancy to a great extent [36] but introduce high computational complexity [35]. Moreover, they require extensive information exchange which may overburden the network with control traffic. While in implicit cache coordination schemes, every content-router does not require pre-information and exchanges only small information with other content-routers. Hence, these techniques can lead towards high content redundancy or may lead to underutilized available space.

However, we argue that all these in-network caching strategies are not optimized because they are formulated only on a single parameter. The performance of implicit cache coordination schemes can be improved if we design caching strategies based on multiple parameters. The decision about content placement is complex and a single attribute is not sufficient for taking intelligent content placement decisions. For this reason, there is a need to select multi-dimensional attributes for taking intelligent content placement decisions. With this in mind, in this chapter, we present a caching scheme, called Multi-Attribute Caching Strategy (MACS) for Content-Centric Networks that is based on using multiple attributes instead of a single attribute. MACS calculates the caching probability by considering content's relevant parameters and network's relevant parameters. The scheme promises to overcome inefficient cache utilization by intelligently selecting caching locations along the content delivery paths.

3.2 MACS Overview

As per default cache policy of CCN, each content router on the downloading path can potentially cache the contents to answer future queries as can be seen in Figure. 3.1(a). However, MACS is based on the principle that using more knowledge about a number of parameters (content and network specific) can help us to improve cache utilization within the network.



(a) Basic CCN Cache Strategy: Copy the content at each CR along the downloading path

(b) MACS Cache Strategy: Copy the content at subset of CR along the downloading path

FIGURE 3.1: The comparison of CCN default strategy and proposed MACS strategy.

Efficient selection of a content router (CR) to cache content along the downloading path helps in improving performance of CCN by keeping content redundancy to a lower level in the network, as shown in Figure. 3.1(b), while offering an acceptable QoE to the user at the same time.

MACS not only takes decisions about storing a content into cache, it also assigns a freshness value to every stored content, which results in automatically removing an expired content to free-up space at a router. With MACS, content freshness is updated on every successful look-up for a stored content and the content is removed from the cache when its freshness expires. This dynamic adaptation of freshness value is dependent on user's requests for the content, and act as a filter to determine the popularity of the stored content. As a result, our scheme keeps the content closer to the region where it is more requested (popular) for longer duration. Furthermore, less popular contents are eventually removed from the cache over a period of time. MACS also introduces a replacement mechanism (Section 3.4.7) to improve cache management, which eventually amplifies the overall network performance.

3.2.1 Basic Mechanism

MACS is based on the principle that using more knowledge about a number of network parameters (content-specific and content-independent) can help us improve cache utilization of content routers while providing acceptable performance to end users. Choosing efficiently which content routers along a downloading path should cache content, can help improve performance of CCN by keeping content redundancy to a lower level in the network. MACS not only includes making decision about at which CR a particular content should be stored, but it also computes and utilizes the content lifetime at each CR that stores the content. For making decision to store a content, we use *content popularity*, *degree of CR*, *storage space* and *distance from the consumer node*. This way, content will be store when following conditions are met:

- Content is most popular
- Node degree is more than two
- Hope count is smaller (or closer to the requesting client)
- Maximum storage space

3.3 MACS Example Scenario

In Figure. 3.2, we present an example to explain the work-flow of content caching in MACS, as a result of a sequence of content requests from clients, while a repository server is accessible to the client via a network of CRs.

Initially, all CRs between clients and server have no cached content. At time t_0, client A sends an *Interest* packet for content C, to the server as shown in Figure. 3.2. In response, the server sends the *Data* packet to A. On the downloading path, each CR calculates its feasibility to store an incoming content and eventually decides whether to cache the *Data* packet or not.

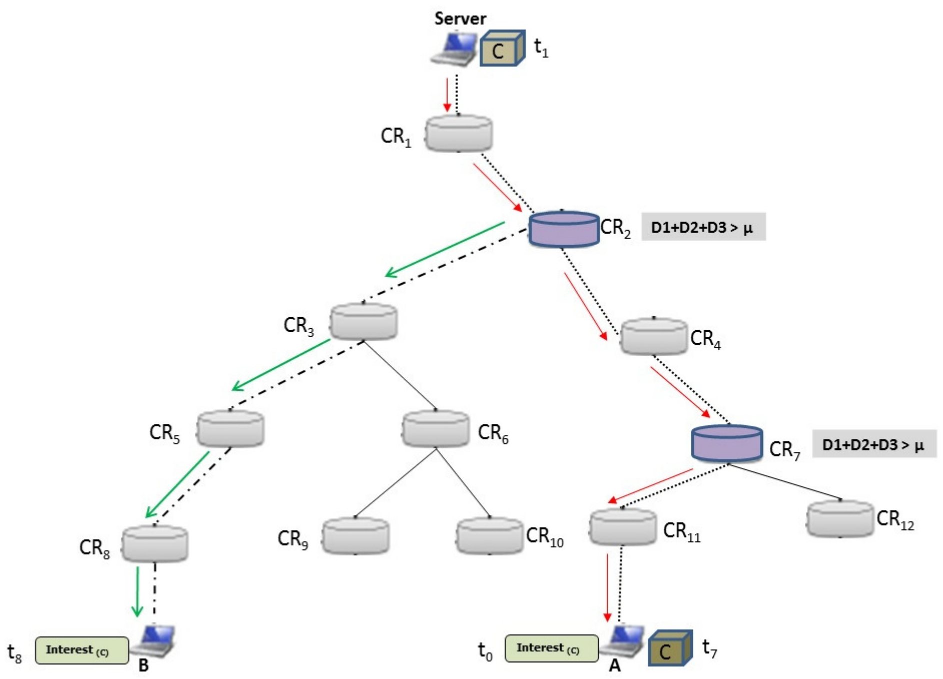


FIGURE 3.2: MACS work flow example

When receiving the *Data* packet, CR_1 calculates its cache decision score value and compare it with a threshold. If calculated value is less than the threshold, it forwards the *Data* packet to the next downstream router without caching. Subsequently, CR_2 receives the content, repeats the process, and caches the *Data* packet if its calculated value is greater than the threshold, while assigning a freshness value to the content. This process is repeated at each CR along the downloading path until content arrives at A. At this moment, CR_2 and CR_7 are only two content-routers that hold the content C because these content-routers have the high cache decision values as compared to other content-routers along the path. Clearly, by caching content on a subset of CR along the path, our proposed MACS scheme is saving network resources such as storage space, number of caching operations and bandwidth as compared to the default CCN caching scheme.

Subsequently, at time t_8, client B sends an *Interest* for the same content C, which is served at CR_2, and as a result, CR_2 also updates the freshness value of the content. This process helps to dynamically determine the popularity of the content within the region around this router.

3.4 MACS Design Details

3.4.1 System Model

We consider a network consisting of R cache-capable routers. We have $C = \{1, 2, ..., ., C_n\}$ content items that can be requested by K users. Following the convention in the literature, we assume that content units are of the same size because of slot size and each cache slot in a cache store can accommodate one content unit at any given time [20]. Let S be the number of content servers. We denote $X = \{x_1, x_2, ..., ..., x_p\}$ the input parameters in the system and relevant importance of each parameter is determined through weights $W = \{w_1, w_2, ..., ...w_p\}$.

In MACS, every node independently calculates the $CD_{Score}(c)$ to store a content. Therefore, we can safely say that MACS supports content level cache. However, the design can be easily extended for chunk level cache where each chunk of a given content is required to be independently stored at a router.

MACS flow model is described in Figure. 3.3. In the following subsections, we explain each component in detail.

3.4.2 Composite Cache Decision Score

In MAC, every downstream content router r∈ R (where R is the number of routers in the network), individually calculates its composite cache's decision score ($CD_{Score}(c)$) to store an incoming content. The content is cached only if the calculated score is above than a defined threshold value μ. Otherwise, the content is forwarded to the next downstream router without caching it. Now, the composite cache's decision score ($CD_{Score}(c)$) at content router r can be calculated as:

$$CD_{Score}(c) = \begin{cases} 1 & if \quad CD_{Score}(c) \geq \mu \\ 0 & Otherwise \end{cases} \tag{3.1}$$

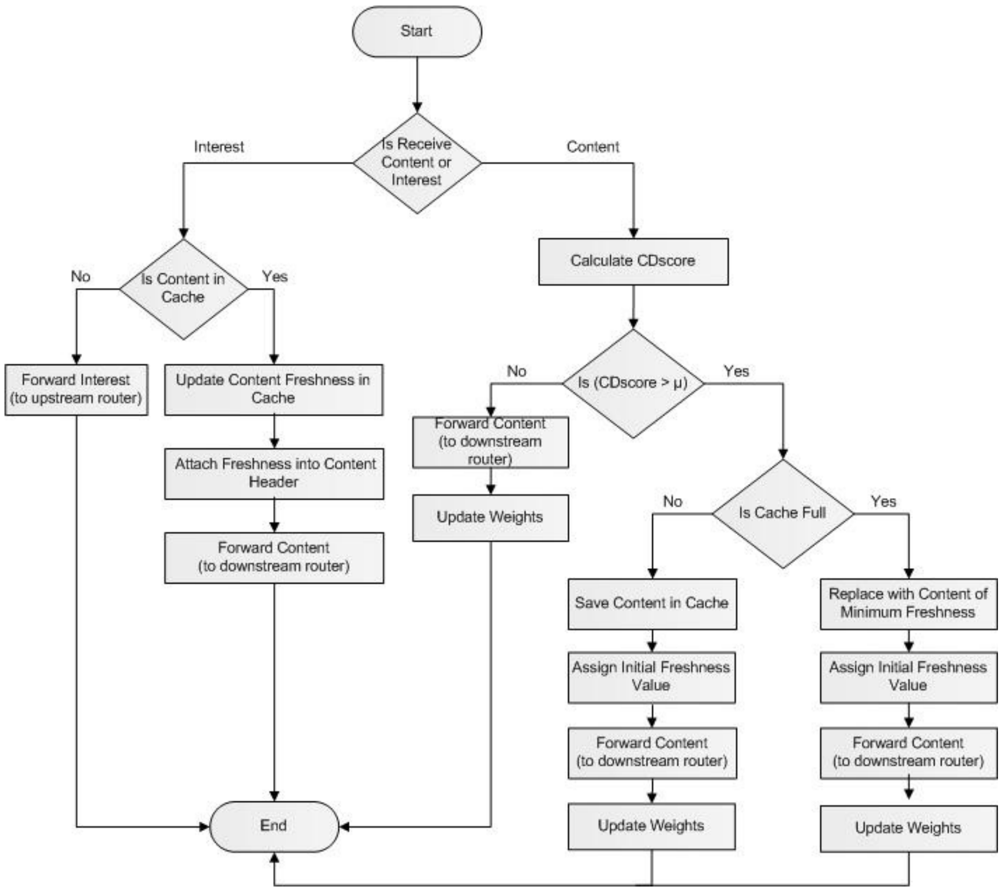

Start
Is Receive Content or Interest
Interest
Content
Is Content in Cache
No
Yes
Forward Interest (to upstream router)
Update Content Freshness in Cache
Attach Freshness into Content Header
Forward Content (to downstream router)
Calculate CDscore
Is (CDscore > μ)
No
Yes
Forward Content (to downstream router)
Update Weights
Is Cache Full
No
Yes
Save Content in Cache
Replace with Content of Minimum Freshness
Assign Initial Freshness Value
Assign Initial Freshness Value
Forward Content (to downstream router)
Forward Content (to downstream router)
Update Weights
Update Weights
End

FIGURE 3.3: Operations of a CR in MACS strategy when receiving a content *Data* and *Interest*

Composite cache decision score is calculated keeping in view a number of parameters which are related to (1) topological characteristics, and (2) content-specific attributes. Some of the most relevant topological parameters used in literature are (1) degree of node [15], (2) distance/hop count [92], (3) available cache space [15], [93], (4) cache size [15], [94], [36]and (5) network size [95]. These are examples of content parametes (1) content size [96], (2) content popularity [48].

The generalized formula to calculate the composite score with MAC while considering multiple dimensions is as follows:

$$CD_{Score}(c) = \sum_{i=1}^{p} (w_i \times x_i) \tag{3.2}$$

The individual score of each parameter is combined in the composite decision score for maximizing the content storing feasibility of a router.

The relevant importance of each parameter is determined through weights w, which are used to optimize the content cache decision. The sum of all weights used for different parameters is always equal to one as presented in Equation. (5.3).

$$\sum_{i=1}^{p} w_i = 1 \tag{3.3}$$

3.4.3 Attributes of MACS Caching Scheme

While the MACS does not rely on specific parameters to function, in the following, we present parameters with justification which we consider to calculate the composite cache decision score. Although, the list of parameters that can be taken is large. However, in order to combine metrics of variable domains (discussed in chapter 2), we considered key parameters related to content, network topology, and router's local characteristics. In following, a detailed discussion on the importance and calculation of selected parameters is presented.

3.4.3.1 Hop Count/Distance

Storing multiple copies of content at all neighboring nodes adds unnecessary redundancy into the network that affects the network performance. Therefore, there is a need to store contents at such places that are at a larger distance from the server or content provider. Subsequently, for cache placement decisions, we consider the distance in terms of number of intermediate hops between the current content provider and the decision-making CR on the returning path.

The distance score $\hbar_{(r)}$ of every content router r∈R on returning path is computed as follows:

$$\hbar_{(r)} = \frac{\Omega_{(s,r)}}{\Omega_{(k,s)}} \tag{3.4}$$

Where $\Omega_{(s,r)}$ is number of hops from the content provider s (i.e. the server) to current content router r. To obtain the distance from the server to the current content router, a hop-count field is added to the *Data* packet. The value of the hop-count field is increased by one each time the packet is forwarded to a downstream router. $\Omega_{(k,s)}$ is the total number of hops from the client k to content provider s (i.e. the server).

3.4.3.2 Content Popularity

Storing popular data at a CR has certain merits in increasing the re-usability of content in terms of a number of hits– within any given region. Inevitably, many techniques (e.g. [44], [52], [59], [93], [97], [98]) consider content popularity as a prime parameter to take cache decisions. If we consider only the strategic location of a CR for caching, there are chances that the cache is filled with less popular contents (i.e. data which is not frequently requested). Therefore, MACS considers content popularity along with other router and network characteristics, which gives preference to frequently requested data over less popular contents while competing for cache. What distinguish our approach in considering content popularity over other techniques is that, in MACS we do not maintain any separate data structure to calculate request count etc.

Moreover, an initial freshness value β is assigned to a stored content and content freshness is updated on every cache hit, the content is removed from the cache after expiry. We defer the discussion on how content freshness is maintained in MACS, Section 3.4.6. However, it is pertinent to mention that in MACS, on every successful lookup, the corresponding CR attaches the freshness value of the content into *Data* header before forwarding to the downstream router.

Assume $N_{(freshness)}$ (i.e. in terms of time) is the new freshness assigned on successful look-up at the upstream router, and $Current_{(time)}$ is the current time. In this way, the updated duration value χ is extracted and as it can be seen in Equation. 3.5 that a high value results in larger popularity value of the content. Let β as default initial freshness value assigned to every newly inserted content as defined in Section 3.4.6 at each node. Then updated duration value χ is derived as follows:

$$\chi = N_{(freshness)} - Current_{(time)} \tag{3.5}$$

Next, we calculate the popularity of the content $\rho_{r(c)}$ at every downstream router r as follows:

$$\rho_{r(c)} = \begin{cases} 0 & if \quad \chi \leq \beta \\ 1 - (\frac{\beta}{\chi}) & Otherwise \end{cases} \tag{3.6}$$

3.4.3.3 Degree of Node

Degree of connectivity of a CR is defined as the number of interfaces attached to the CR. Usually, majority of the network traffic flows through the highly connected points/routers. Therefore, selecting routers with a high degree of connectivity to store contents can make the content more visible as the CR is approachable from different directions or parts of the network. A large number of caching techniques [15], [17], [36] utilize global knowledge about network topology or information regarding physical location of a node while taking cache decisions in which the node degree is the prime parameter. A node with high degree is considered a better option to cache the content, as the node can serve more neighboring clients simultaneously.

The main advantage of this parameter is that it reduces the cache redundancy by storing the content in the prime location. The cache performance of the system can also be optimized through this parameter.

The degree score of a node r on a forwarding path from the content provider to client is defined as follows:

$$\partial_{(r)} = \frac{d_{(r)}}{d_{max}(s, k)} \tag{3.7}$$

Where $d_{(r)}$ is the degree of node r∈R and $d_{max}(s, k)$ is the highest available degree of any node on the downloading path from the content provider s (i.e. the server) to a client k.

3.4.3.4 Available Cache Space

Cache space is very important in CCN, as caching is an inherent feature of the architecture [15]. Therefore, it is important to use the available cache space at CR's in an efficient way.

For content storing, picking only the strategically important routers on the returning path can cause congestion. On the contrary, the resources that are present on less strategically important areas are usually under-utilized cache resources. This load imbalance creates congested data points at the CR with the high score to store incoming contents, and hence it may lead the network into a sub-optimal state. Frequently occurring content replacements due to congestion may result in increasing the delay experienced by the end-users. However, in MACS, we suggest distributing the overhead over network routers for a better utilization of available network capacity.

Available cache space $CS(av_{(r)})$ of $CR_{(r)}$ is calculated according to Equation. 5.5.

$$CS(av_{(r)}) = \frac{CacheSize_{(r)} - CacheSize_{(r)used}}{CacheSize_{(r)}} \tag{3.8}$$

Where, $CacheSize_{(r)used}$ represents the used cache space on $CR_{(r)}$ and $CacheSize(r)$ is the total cache size of $CR_{(r)}$.

3.4.4 Threshold Analysis

A threshold value is used as a filter against the content decision score value, in order to make a decision about whether to cache the content or not. Inevitably, defining an appropriate threshold value is vital to improving the performance of the network. If the threshold value is too low, a router can rapidly get congested, while a very high value may lead to under-utilization of network resources, eventually degrading performance.

We have computed the cache hit ratio against the threshold values, as shown in Figure. 3.4, we can see that the performance is highly affected by a very large value of threshold (that is 1), as contents are only saved on few routers of higher score.

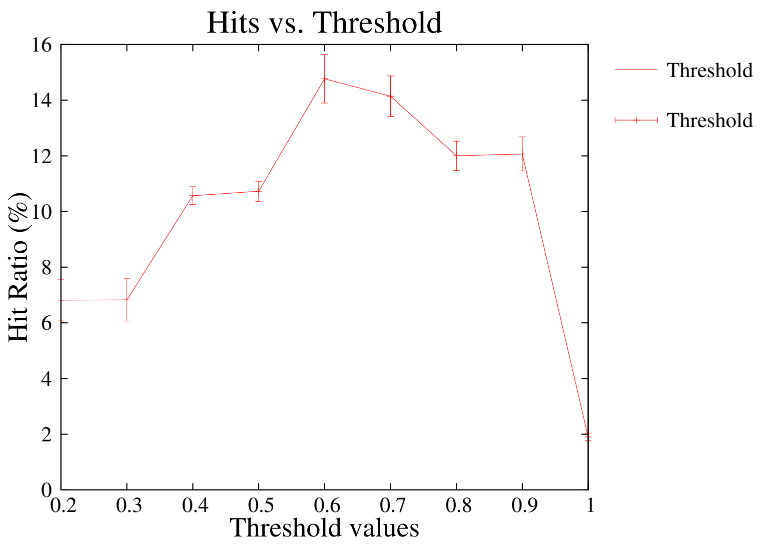


FIGURE 3.4: Hit ratio at different threshold values μ

3.4.5 Weight Assignment

With multiple attributes contributing towards the weighted cache decision score, the default strategy of MACS is to uniformly assign the weights among all contributing attributes. The results of this strategy were presented in [99]. However, we argue that the performance of MACS can be greatly improved by considering relative importance of the attributes, which suggests that the weights are adaptive and are computed accordingly to the overall network condition. For instance, a CR's strategic importance relative to different clients, content popularity and available capacity to store different contents vary over time and cannot be easily predicted.

Hence, we propose a solution that is capable of learning these dynamic patterns. This helps in building an optimized combination of selected attributes to maximize a CR's score for saving contents at any given time instance. In other words, we suggest the need to dynamically define the relative importance of the considered dimensions which are individually calculated at each CR along the forwarding path.

In MACS we employ error correction learning for dynamic weight adjustments, which is a well-known learning technique of neural networks [19]. Let us assume that the desired score value is 1, we can then define the error function δ at $CR_(r)$ as follows:

$$\delta(r) = (1 - CD_{Score}(c)) \tag{3.9}$$

The error is then back-propagated as an input to $AdjustWeight$ function which is defined in Algorithm-1 (at table. 3.1).

TABLE 3.1: Weight adaptation at CR r

ALGORITHM-1

```
1.  $AdjustWeight\ (\delta(r))$ {
2.    $\forall i \in (1...p)$
3.       $\Delta w_i = l * \delta(r) * X_i$
4.       $w_{i(new)} = \eta * w_{i(old)} + \Delta w_i$
    // To normalize weights to have their sum equal to one
    as following [100]
5.    $\forall i \in (1...p)$
6.       $w_i = \frac{w_{i(new)}}{\sum_{j=1}^{p} w_{j(new)}}$ }
```

In Algorithm-1 (line 3), X_i is the corresponding i^{th} parameter, such as, CR degree or cache space for which the Δw_i is calculated. Here, l is the learning rate, which determines the pace of adaptation. With the high value of l, the resulting Δw_i at each iteration is large. Whereas, if we select a small l, the weights will be adjusted slowly. The Δw_i of the i^{th} parameter is derived by multiplying the corresponding value with the error and learning rate (line 3). Subsequently, the new weight is calculated by adding this delta change into old weight value (line 4) as defined in [101]. η (line 4) is the momentum parameter [101] that helps to improve convergence time of the algorithm. The default value of η is taken as 0.9 [102]. As already discussed in Section 3.4.2, the sum of all the corresponding weights must be equal to 1. Therefore, after every update, weights are again normalized to limit the values within the range $[0-1]$ $\forall i \in (1...p)$ (line 6). In

TABLE 3.2: Pseudocode of Weight adaptation

```
PROGRAM AdjustWeight:
    Read Error Function $\delta(r)$;
    Read Parameter X$_i$;
    FOR(i=1; i $\leq p; i++$)
        $\Delta w_i = l * \delta(r) * X_i$;
        $w_{i(new)} = \eta * w_{i(old)} + \Delta w_i$;
        Print w$_{i(new)}$;
    ENDFOR;
END;
```

order to maximize its score independently, each node adapts the corresponding weights for playing an effective role within the connected area. The pseudo-code of weight adaptation is defined in Table. 3.2.

We can extrapolate the effectiveness of dynamic weight adaptation using a topology which is shown in Figure. 3.5. To demonstrate the weight adaptation process, we show the behavior of weight adaptation at *node-9* and *node-16* in Figure. 3.6, which are present in different strategic locations, thereby highlighting the ability of our algorithm to adapt itself according to the network condition and neighborhood. From Figure. 3.6(a), it is apparent that weight of distance and degree is high because node-16 is directly connected to a large number of clients, whereas the weight of the two other parameters, storage space and popularity are low because at this node, number of connected clients are large and maximum storage space is utilized.

Subsequently, as shown in Figure. 3.6(b), node-9 has small degree weight but high storage space weight because it is connected only to a single client and it can possibly store only one client's requested data.

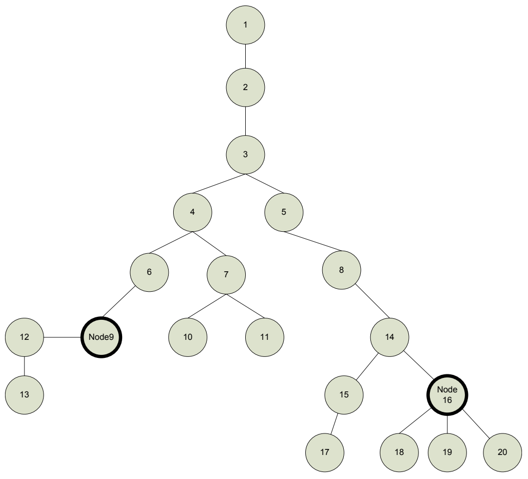
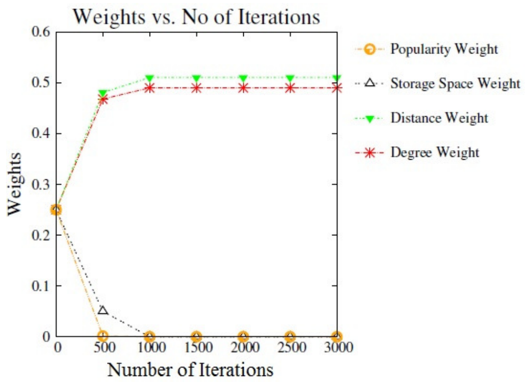
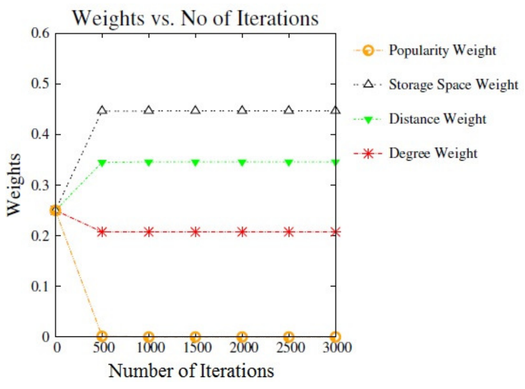


FIGURE 3.5: A tree network scenario for weight adaptation at Node-9 and Node-16



(a) Weights adaption at Node 16



(b) Weights adaption at Node 9

FIGURE 3.6: Weights on two different nodes

3.4.6 Content Freshness

In MACS, a content lifetime/freshness is assigned to every stored content in CR, which is a function of time. This freshness determines the amount of duration for which the content is available at a router to answer future queries of the same. The freshness of content decreases over time and content is removed from the cache after expiry. The freshness helps in effectively utilizing the limited available in-network space based on current user's interests.

On every successful look-up for a content at an in-network content-provider, an amount of default freshness value is added to the old freshness values of the content that is shown in Table. 3.3. Increasing freshness on every successful lookup has

TABLE 3.3: Content-Freshness calculation at CR 'r'

ALGORITHM-2: Receiving *Interest* Packet

```
1. Let $\beta$ is the initial freshness value
2.  IF Content in Cache then
3.    $LT_{old} = GetLT(Content)$
4.    $LT_{new} = LT_{old} + \beta$
5.    $CacheUpdate(LT_{new}, Content)$;
6.  Forward(Content);
7.  Else
8.  Forward(Request);
```

Receiving *Data* Packet

```
1. Calculate $CD_{Score}$ at router $r$
2.  IF $CD_{Score}(c) > \mu$ then
3.    Cache($\beta$, Content);
4.  Forward(Content);
```

the potential to dynamically filter out popular contents from less popular data. A least frequently requested content data is removed from the cache after expiry to free-up the space at the given CR. On the other hand, a higher freshness value attached to popular data help in keeping the latter for a longer duration to address future requests. The pseudo-code of content freshness calculation at CR_r when it receives an *Interest* packet is given in Table. 3.4. Although the concept of assigning content lifetime is not new, in [103] consumer driven freshness is used to determine lifetime of data. In this, consumers request a particular freshness of data, if it is present on intermediate routers, request is served. Otherwise its is request from source node. Our approach is very similar to [52] paper content lifetime is assigned to every piece of content before it is being cached on any router. However, it is taken content lifetime as a static value based on predefined weights. Our work differentiates from this work since MACS does not rely on the content static freshness value, as well does not exploit predefined weights. The strategy we present, determines the relative importance of each dimension with a dynamic weight adaptation mechanism.

Table 3.4: Pseudocode of Content-Freshness

```
PROGRAM ContentFreshness:
    Read Packet;
    Read Initial Freshness Value $\beta$;
    IF (Interest Packet)
        IF (Content in Cache)
            Update Content Freshness in Cache;
            Attach Freshness into Content Header;
            Forward Content Packet to Downstream Router;
        Else
            Forward Request Packet to Upstream Router;
        ENDIF;
    Else
        At Current Router Calculate $CD_{Score}(c)$;
            IF ($CD_{Score}(c) > \mu$)
                Cache Content with Initial Value $\beta$;
                Forward Content Packet to Downstream Router;
            Else
                Forward Content Packet to Downstream Router;
            ENDIF;
    ENDIF;
```

3.4.7 Cache Replacement

MACS is equipped with new cache replacement mechanism as follows: when we store the content on any router, an associated freshness value is attached with it and, this value is decreased over time. On every lookup, the freshness value is also increased. A low freshness value indicates that the content can be safely replaced without affecting the performance. This associated freshness of currently stored contents is also used in MACS while taking replacement decisions. When a new content comes then its score value is compared to the defined threshold, if it is greater than the threshold then the content will be stored. However, if the current cache is filled (line 4) then the content that has smallest freshness value will be replaced with the new content (line 5-7). In this way, the replacement technique of MACS tends to favors the most popular contents. The pseudo-code of content replacement calculation at CR_r when it receives a *content* is given in Table 3.5.

TABLE 3.5: Pseudocode of Cache-Replacement at CR 'r'

ALGORITHM-3

```
1. Let set $S_{cr}$ contains all the stored content at a given
   router $r$ along with the freshness values
2. Let $M_r$ is defined as the maximum size of cache at
   router $r$
3. For every incoming content 'c' to be stored in cache
4.   IF ($| S_{cr} |= M_r$) Then
5.     $S_{cr} \leftarrow S_{cr}$ - $\{c_y \in S_{cr}$ | where y has minimum
       freshness value
6.     c $\leftarrow \beta$ (where $\beta$ is the default initial freshness)
7.     $S_{cr} \leftarrow S_{cr} \cup$ c
8.   EndIF
```

3.4.8 Complexity Analysis of MACS

In MACS, cache decisions are taken by considering the individual score of multiple parameters. As we use additive property, time complexity of scores estimation is equal to the highest time complexity of calculating any individual score. The parameters we use in this thesis such as degree, hop count and free space etc., scores have constant time complexity. Hence, time complexity of scores estimation in MACS of constant time i.e. $\mathcal{O}(1)$. Time complexity of weight adaptation in MACS is $\mathcal{O}(n)$ for n attributes hence we conclude this total time complexity for taking a decision is $\mathcal{O}(n)$.

In our proposed scheme, evaluations are made on most recent available information from control/data packets such as maximum path degree and hop count. There is no need of extra space at the node to keep record of information. Hence, space complexity of MACS is $\mathcal{O}(n)$ where n is the total number of attributes.

3.4.9 Conclusion

In this Chapter, we present a cache management scheme MACS to support intelligent content placement decisions in CCN. In MACS we consider multiple parameters leveraging the advantage of both content and topological characteristics of the given network. We also demonstrate that error correction learning techniques from neural networks can be effectively applied to determine relative importance of each attribute within a particular scenario. Furthermore, in order to effectively utilize the limited available in-network space, we dynamically update freshness of a content. For this purpose, in this chapter, a new cache replacement policy that is based on freshness value of a content is proposed. This scheme promises to overcome inefficient cache utilization by intelligently selecting caching locations along the content delivery paths in order to increase chances of cache hits.

Chapter 4

Simulation and Performance Evaluation

We demonstrate the performance of MACS by evaluating it on a number of performance metrics including Cache Hit Ratio, Hop Reduction Ratio, Content Reusability and Used Buffer Ratio and showcase their behavior as a function of parameters such as Content Distribution, Cache to Population Ratio and Time. There are many parameters like server load, Round-trip Hop Distance etc. However, above mentioned parameters are the most commonly used metrics to evaluate the techniques [54], [18], [48]. Moreover, we compare the performance of MACS against well-known schemes such as

- Age-based Cooperative Cache Scheme (ABC) [53]
- Optimal Cache Placement strategy based on Content Popularity (OCPCP) [59]

In ABC, content ages is assigned to every content while in OCPCP, cache decisions are based on content popularity.

4.1 Simulation Environment

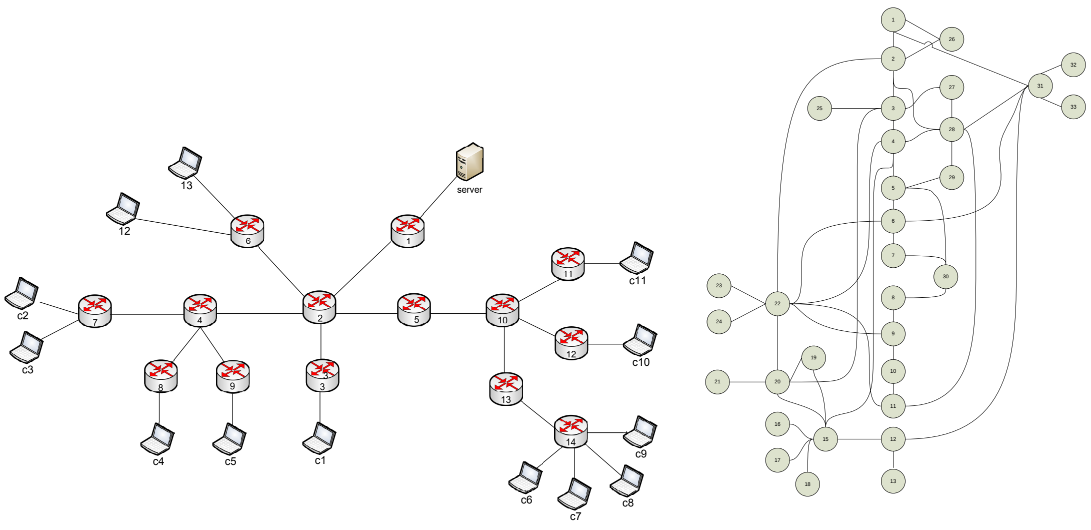
We use the open-source ndnSIM simulator [20] that implements NDN protocol stack for NS-3 network simulator. In order to evaluate MACS with different topologies, we use the real network typology GEANT [104] and a tree topology, as shown in Figure.4.1. GEANT is the pan-European data network that interconnects European research and educational community and it is composed by 22 content-routers. We distribute the endpoints (i.e. server and consumers) randomly and attach a variable number of consumer nodes ($1 \sim 3$) to each randomly selected content-routers. Our topology contains 10 consumers and one server.

In our simulations, each content provider serves 100 different content objects. The uniform size of cache (CS) is 50 and the best route policy [20] is used in the network. As flooding strategy is accompanied with its own extra overheads across the network, so, in order to have a fair comparison we prefer Shortest Path Routing. Zipf(α) is implemented as a content popularity distribution that is simply tuned by a single parameter α. Content request q is modeled as Poisson processes. We have chosen the threshold value as 0.6 (Figure 3.4) and a discussion on choosing threshold value is discussed in Section 3.4.4. Simulation results are taken by running each scenario ten times with different seed values and for each caching strategy; furthermore, we report the average values with the 95% confidence intervals. Table. 4.1 lists the performance parameters which use for the evaluation [59], [105], [106].

4.1.1 Performance Metrics

We assess the caching schemes considering the following four aspects: Network Cache Hit Ratio, Hop Reduction Ratio, Content Reusability, Used Buffer Ratio.

Network Cache Hit Ratio: Cache hit ratio is a strong measure of load reduction over server due to in-network cache hits, and it is defined as follows:



(a) Tree topology (b) The Geant network topology

FIGURE 4.1: Simulation topologies

TABLE 4.1: Simulation parameters

Attribute	**Value**
Typologies	GEANT, Tree
Number of different content	100
Content size	1024B
Cache size (number of contents)	10 $\sim$ 50
Link delay	10ms
Bandwidth	10Mbps
Interest frequency	3/s
Simulation time	200s
Simulation runs	10
Threshold μ value	0.6
Initial freshness value (i.e. β)	100s

$$\psi = \frac{\sum_{i=1}^{N} Hits(i)}{\sum_{i=1}^{N} Req(i)} \tag{4.1}$$

Where, $Hits(i)$ represents the total number of hits and $Req_{(i)}$ represents the total number of received requests on a content router i.

Hop Reduction Ratio: Hop reduction ratio signifies the impact of delay, an in-network cache scheme offers, and it is calculated as follows:

$$\gamma = 1 - \frac{\sum_{r=1}^{R} h_r}{\sum_{r=1}^{R} H_r} \tag{4.2}$$

Where, h_r is the hop count from the client to the content provider (which serves the request) and H_r is the hop count from the client to the original server. A high value of hop reduction ratio means less end-to-end delay experience by the users and vice versa.

Content Re-usability: We define content re-usability as the average number of times saved contents on an intermediate CR are re-used (request served by in-network cache) against the occupied space within the given network. It is defined as follows:

$$\xi = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{CacheHits(i)}{OccupiedBuff(i) + 1} \right) \tag{4.3}$$

Where, $CacheHits_{(i)}$ represents the total number of cache hits on the $CR_{(i)}$, and $OccupiedBuffer_{(i)}$ represents used buffer space on $CR_{(i)}$, and N is the number of CRs within the network. A high average re-usability supports the effectiveness of cache decisions made by any given in-network caching approach.

Used Buffer Ratio: Used buffer ratio measures the amount of in-network space filled with contents to serve future requests against total available space of a network. It directly reflects the efficiency of a scheme with respect to resource management, and it is defined as follows:

$$\Phi = \frac{\sum_{i=1}^{N} CacheSize(i)_{Used}}{\sum_{i=1}^{N} CacheSize(i)} \tag{4.4}$$

Where, $CacheSize(i)_{used}$ represents the used cache space, $CacheSize_{(i)}$ is the total cache size of CR i, and N is the total number of intermediate content routers within the given network.

4.1.1.1 Scenario 1 : Comparison Against Varying Content Popularity Distribution on Tree Topology

The aforementioned metrics are measured on the tree topology, as shown in Figure. 4.1(a). The tree-based topology consists of 13 consumers, 14 CR, 5 average node degree and a single server to satisfy the requested interests. A large number of consumer taken to identify the importance of efficient in-network cache for server load. Ten runs are taking different seeds values. Here, we assume that every intermediate router has the same cache size (i.e. 10) and content population size is fixed to 100 contents, and they follow a Zipf popularity distribution [107], characterized by the skewness parameter α. In the Zipf-Mandelbrot law distributions, the occurrence rate of incoming content of the k^{th} rank is proportional to $1/k$. Content popularity distribution determines the diversity of contents that can be requested from different clients. A low value of α means that the diversity of requested data is high in the network, while a higher value of α suggests that the probability of requesting the same kind of contents is high.

In Figure. 4.2, the value of α increases on the x-axis to compare the behavior of a given scheme against varying content popularity distribution. In particular, α is set in the range $[0.3 \sim 1.8]$ and requests are generated randomly from all nodes according to the Zipf-distribution. In this way we explored the impact of request frequency distribution on the performance of our proposed caching scheme.

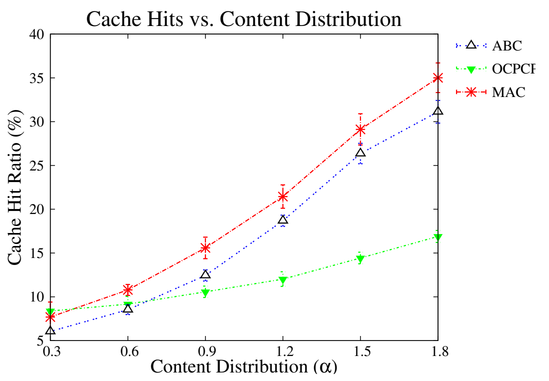
As we increase content popularity, the chance that a stored content is frequently requested within the network also increases. Therefore, an increase in cache hit rate of the network is observed with increasing content distribution in Figure. 4.2(a). We can see that MACS has a better cache hit rate than ABC but with a small amount of buffer usage as shown in Figure. 4.2(b). In the given tree topology, nodes process a high degree of connectivity within the network. Secondly, high diversity in the content population with resource limitation at intermediate content router contribute in filling the network with a large number of contents for answering future requests. Therefore, we see that buffer usage of all the schemes, in Figure. 4.2(b). However, MACS is able to maintain a significant low buffer usage as compared to other schemes, and it can further decreases with increasing time in case of MACS. This is due to decrease in diversity of content requests from different end users (clients). Nevertheless, buffer usage remains the same in the case of OCPCP and ABC, as these schemes fail to adapt to changing network scenarios as compared to MACS. In MACS, a strategically important location is favored to store content in order to increase cache hit rate of the network, (Figure. 4.2(a)) and to optimize available storage space (Figure. 4.2(b)). Therefore, we can observe a small hop reduction ratio with MACS in Figure. 4.2(c) as compared to other schemes, but the gap tends to decrease at increasing content distribution with small buffer usage and increasing hit rate and content reusability Figure. 4.2(d). We can say that MACS provides a better trade-off in terms of hop reduction ratio against compared schemes. Hence, we can conclude that carefully storing content within the region, where it is most popular helps in delivering satisfactory performance to end users in the resource-stringent environment.

4.1.1.2 Scenario 2 : Comparison Against Cache to Population Ratio on GEANT Topology

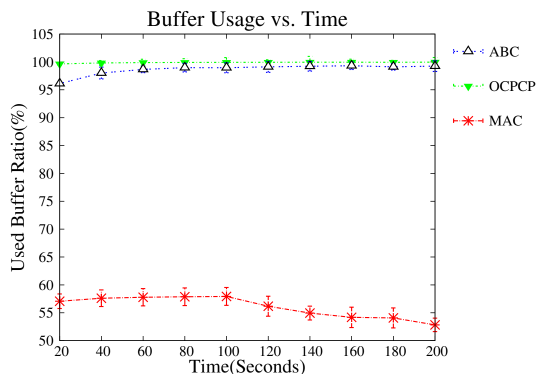
In Figure. 4.3, a comparison of MACS shows with other cache schemes against increasing cache to population ratio -on GEANT network topology (Figure. 4.1(b)) as described in Section 4.1. In this, we take two servers and ℓ as the fixed population of contents available on servers- that can be requested by different clients within the given simulation time; $\Im$ as the cache size (i.e. in terms of content units) of any given intermediate router. Then cache to population ratio can be defined as follows:

$$C_{pr} = \frac{\Im}{\ell} \tag{4.5}$$

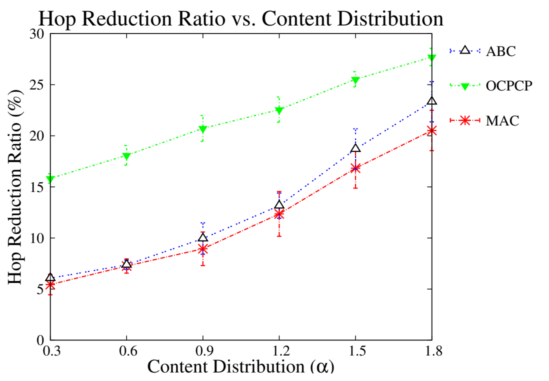
For simplicity, we assume that every intermediate router is configured to have varying cache size $\Im \in [10, 60]$ in the given scenario shown in Figure. 4.3. Here, content distribution value α is set to 0.3, and content population size is set to 5 x 10^2 contents. We believe that cache to population ratio strongly characterizes the in-network cache management challenge. It also provides an approximate measure



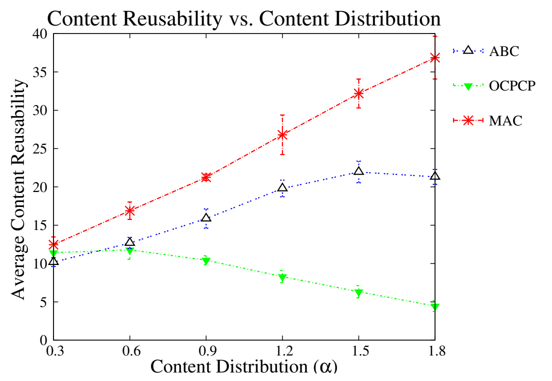
(a) Analysis of cache hit ratio as the function of varying content popularity distribution



(b) Analysis of average buffer usage as the function of time



(c) Analysis of hop reduction ratio as the function of varying content popularity distribution



(d) Analysis of content reusability as the function of varying content popularity distribution

Figure 4.2: Instantaneous behavior of the caching schemes for a Tree topology with different Zipf α

of the efficiency of the given scheme to fulfill user's demands in the best possible way with varying network resources.

We can see in Figure. 4.3(a) that cache hit ratio of MACS is higher than all other schemes, and it keeps on increasing with increasing capacity of the network to store a large number of contents at intermediate routers. However, we can see in Figure. 4.3(b) that MACS does not greedily try to occupy network resources and maintains its average buffer usage to a significantly lower limit as compared to OCPCP and ABC schemes.

In OCPCP, each incoming content is initially saved on every intermediate router. The contents are popular then gradually shifted towards the edge of the network that is near to clients. The ability to greedily occupy the available resources helps to maintain small delays, as a large percentage of cache hits usually occur at edge routers. However, in MACS the tendency to effectively utilize the available resources comes with a slight compromise on hop-reduction ratio Figure. 4.3(c).

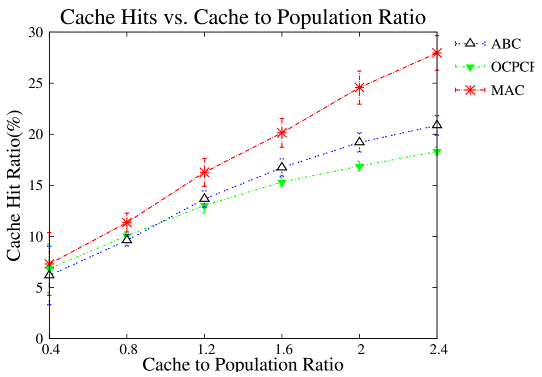
Cache content reusability decreases as cache size increases, OCPCP and ABC tend to occupy all the available resources in the network without taking into account user's interests within the given vicinity. Therefore, with increasing space, the ratio of the amount of reused data with respect to occupied space also keeps on decreasing.

We can see in Figure. 4.3(d) that MACS is able to maintain a higher average content re-usability than OCPCP and ABC. However, it also decreases with increasing cache to population ratio, as due to increasing the capacity of the network to store more data nodes that are not placed at strategically important location can also be favored to store data due to large available free space. These nodes are not connected to a large number of clients. Hence, it impacts the overall content reusability of the whole network.

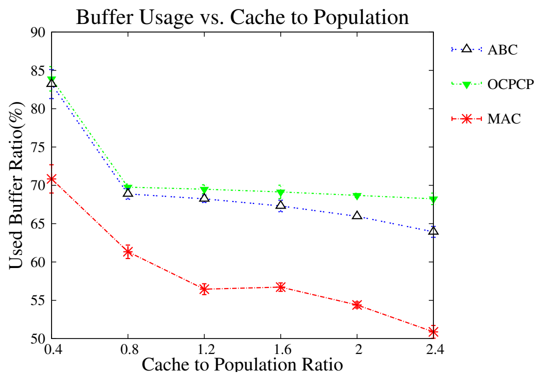
4.1.1.3 Scenario 3: Stabilization Against Time on GEANT Topology

In order to strengthen our findings, we proceed to evaluate the cache performance of MACS with other cache management schemes with respect to the simulation time. In this scenario, we specifically collect the results against time to evaluate; how is our proposed in-network MACS caching scheme stabilizes against time.

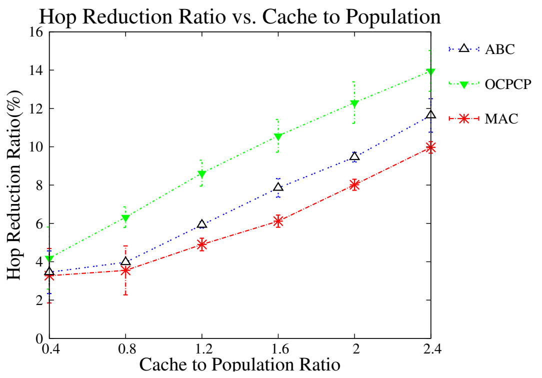
The behavior of MACS along with OCPCP and ABC schemes is shown over time in Figure. 4.4 –on GEANT topology (Figure.4.1(b)) with 2 servers–when the cache to population ratio is set to 1.6. We can see in Figure. 4.4(b) that initial storing rate of incoming contents in MACS at content routers is high due to available large free buffer space. However, we can observe a restricted increase in buffer usage over time with MACS despite available space, as the protocol stabilize itself by learning the environment and user interest patterns around the given content router.



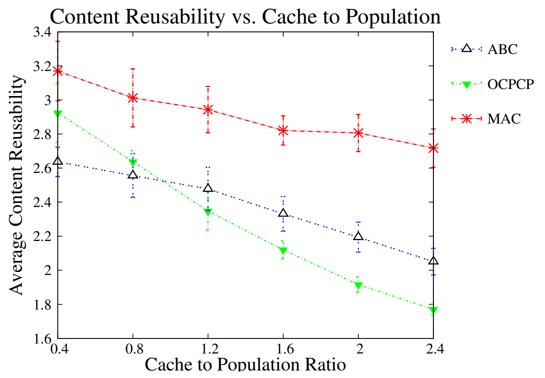
(a) Analysis of cache hits as the function of changing values of cache to population ratio ($C_{pr} = \frac{\Im}{\ell}$)



(b) Analysis of average buffer usage as the function of changing values of cache to population ratio ($C_{pr} = \frac{\Im}{\ell}$)



(c) Analysis of hop reduction ratio as the function of changing values of cache to population ratio ($C_{pr} = \frac{\Im}{\ell}$)

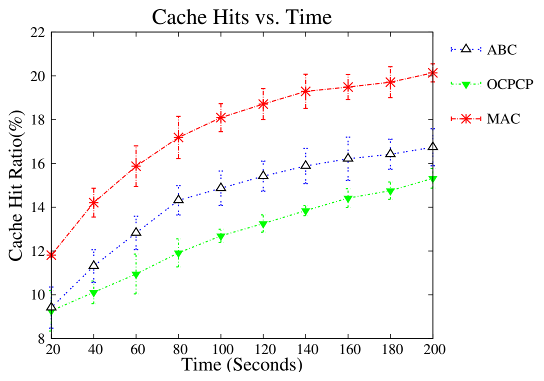


(d) Analysis of content reusability as the function of changing values of cache to population ratio ($C_{pr} = \frac{\Im}{\ell}$)

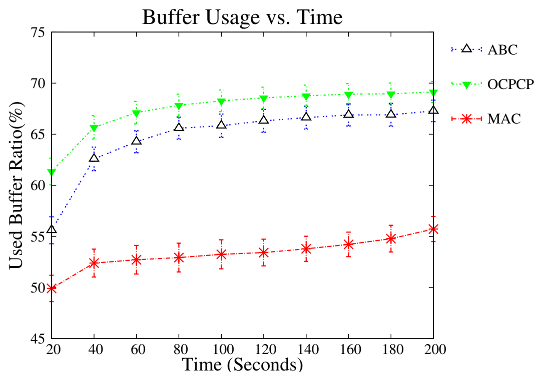
FIGURE 4.3: Instantaneous behavior of the caching schemes for a real Geant topology with different cache sizes

We can see that despite storing a limited amount of contents at different in-network content routers, a cache-hit rate in Figure. 4.4(a), and content reusability in Figure. 4.4(d) in the case of MACS increases greatly than OCPCP and ABC over time. Due to dynamically increase the lifetime of frequently requested contents MACS is better able to prioritize data based on user's interests. The contents that are initially saved but not requested frequently from users around the content router are automatically expired without overloading the network. On the other hand, in OCPCP and ABC the cache space at a given router is blindly filled with incoming contents. This helps in getting an increased performance at the start of the network, but we can see that the two schemes poorly adapt to given network scenario as compared to MACS over time.

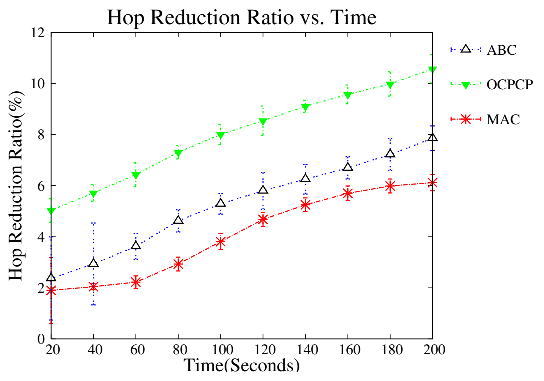
The dynamic weight adaptation and determination of individual score of different



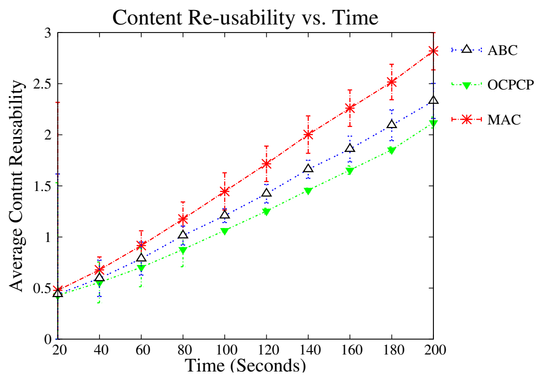
(a) Analysis of cache hits as the function of varying values of time



(b) Analysis of average buffer usage as the function of varying values of time



(c) Analysis of hop reduction ratio as the function of varying values of time



(d) Analysis of content re-usability as the function of varying values of time

FIGURE 4.4: Instantaneous behavior of the caching schemes for a real Geant topology with time

considered parameters in MACS, such as distance/hop-count, also help in storing contents towards the edge of the network. Therefore, a hop-reduction ratio in Figure. 4.4(c) increases over time with MACS.

4.2 Conclusion

In this Chapter, MACS is thoroughly evaluated against a variety of parameters in heterogeneous synthetic and real network topologies. The simulation results showed that cache decision score that is based on multiple attributes can help in intelligent decisions regarding content cache placement. Furthermore, a comparison against well-known caching schemes is also included to depict a more realistic picture on performance gain that can be achieved through MACS.

Evaluations in a variety of heterogeneous environments showed that MACS attempts to overcome inefficient cache utilization and reduces cache load at each node. However, in MACS the tendency to effectively utilize the available resources comes with a slight compromise on hop-reduction ratio. Hence, MACS ability to carefully store a content, where it is most popular helps in delivering satisfactory performance in resource-stringent environments.

Chapter 5

Mobility-based Caching Strategy for Contnet-Centric Networking-Design Details

5.1 Introduction

In areas, where communication infrastructure is not available, and wired networking is costly Mobile Ad-hoc Networks (MANET) can be deployed [108]. MANET can also be deployed in shopping complexes/business areas, where people want to exchange data regarding current fashion trends, price or promotional advertisements. Another useful implementation is in conference meetings, where attendees want to share information about current research trends.

CCN provides a receiver-driven, secure and simplistic model equipped with ubiquitous in-network caching [109], name-based routing [11] and security [14]. In-network caching is used to increase content availability within the network, as it avoids the need to send interest packets for the same content towards the source node. This also results in reducing delays and server load, hence, user's Quality of Experience (QoE) is increased. Network partitioning within MANET is frequent due to mobility; in such environments, mobile users may experience long delays to access data from source. Therefore, mobility greatly effects data availability in these networks, as network capacity is degraded by multi-hop communication whenever network partitioning occurs [110]. In this regard, a content-centric mobile ad-hoc network may get benefit from CCN's in-network caching to improve availability of data within a connected region.

Default architecture of CCN does not provide explicit mechanisms for cache decision; all nodes on the path may cache all incoming contents. This increases data accessibility, decreases retrieval latency and redundant requests sent to server. However, resource inefficiency of this approach makes it unsuitable for MANET. In MANET, data caching strategies are required to cope with resource constraints, limited bandwidth and battery power. Hence, cache placement in mobile networks is a challenging task. It further gets significant at the time of network partitioning, as because of mobility, disconnections occur between nodes due to which nodes within the same network become unreachable to each other.

A lot of research has been carried in recent years to improve caching gain in CCN-based ad-hoc networks. For instance, Yuguang et al. [62], investigate the impact of cache strategy on delivering named data in MANET. In this, few nodes are selected to cache incoming contents on returning path for reducing overhead and data redundancy in MANET. Location-aware caching schemes [64] [65] use location coordinates of source and consumer nodes for forwarding *Interest* and *Data* packets. By exploiting this information in [64], the proposed caching scheme considers distance and remaining energy of a node for caching decision. Probabilistic caching strategies [66] [67] dynamically calculate caching probability at each node for taking cache decisions. This caching probability is calculated by considering content and node related parameters such as, freshness of content, battery level and used cache space of a node. Cooperative caching for MANET [68], [69], [70] is considered widely in recent years. This allows sharing and coordination between multiple caching nodes for efficient utilization of available resources. However, few techniques in literature consider mobility issues [90] for taking cache decisions. For example, Wei et al. [90] propose a mobility and popularity-based caching strategy (MPCS) that takes advantage of user mobility patterns and content popularity for maximizing cache hit rate within the network. Bernardini et al. [78] propose a social based CCN caching strategy. It is based on the fact that users in same social community might be interested in the same content. This cache strategy makes use of understanding regarding how users interact with each other.

None of the aforementioned schemes consider availability of data at the point of network partitioning. We argue that it is an important factor in MANET. When mobile nodes move frequently, network partitioning may occur by dividing the network into disconnected regions. This may lead to unavailability of data within the network, if the provider moves away without replicating cached contents on another node. Furthermore, MANET are known to have constraints on resources like bandwidth, cache space, computational power, and energy. Thereby, it is vital to optimize, 1) what contents to cache, and 2) where to cache by considering node mobility, content diversity and content popularity, for improving network performance. Additionally, it is significant to devise a mechanism which may assist cache schemes for replicating cached contents before partitioning.

In this thesis, we propose a strategy (*Caching Strategy in CCN-MANET (CSCM)*) to cache important contents in such networks. It dynamically adapts its caching decisions for individual contents. CSCM is a generic framework which might be accompanied with existing approaches for improved performance in mobile environments. The proposed scheme is divided into two phases. The first phase is to select optimal nodes for storing incoming contents. The purpose of selecting optimal cache nodes is to reduce redundancy, and to place the data where it is most popular. For this, we adapt Multi-attribute Caching Strategy (MACS) [99] for mobile environments. MACS determines suitable caching locations along the content delivery paths while taking into account multi-dimensional analysis. Second phase selects a new dominator for relocating cached contents, if the previous dominator moves to a disconnected region.

5.2 A Caching Strategy in CCN-MANET

In this section, we elaborate design details of our proposed caching scheme called Caching Strategy in CCN-MANET (CSCM). We can divide CSCM into two parts, 1) cache node selection with MACS [99], 2) popularity and freshness-driven mobility adaptive cache relocation algorithm. On delivery path, CSCM selects optimal nodes to cache incoming contents for answering future requests. The selection is based on multi-dimensional analyses of a node's candidacy to maximize hit rate within the network. CSCM takes into account node centrality, available cache space and remaining energy level for deciding caching and relocation policy. A mobile node is selected for caching when its cache decision value is greater than a defined threshold. Additionally, mobility adaptive cache relocation algorithm selects a new candidate cache node in order to relocate cached contents if the previously selected node moves to another location. Contents selection for relocation considers popularity and freshness of information to improve network performance.

5.2.1 Network Model

We consider an ad-hoc network comprises of a group of mobile nodes, which act as producer, consumer and relays. Relay nodes are interchangeably stated as relays, content routers or nodes throughout the paper. Each node has an omni-directional antenna with same transmission range. Network topology is represented as an undirected graph $G = (V, E)$, where V is the set of mobile nodes $\{r_1, r_2, ..., r_n\}$ and $E \subseteq (V \times V)$ is the set of links between nodes. We denote $X = \{x_1, x_2, ..., ..., ..., x_p\}$ as the set of attributes taken as input for taking cache decisions, and relevant importance of each attribute is determined through weights $W = \{w_1, w_2, ..., ...w_p\}$. One-hop neighbors are defined as nodes which are present within the transmission range of each other. We assume nodes or people carrying wireless devices can freely move around the given region. This may create partitioning within the network resulting in partial disconnectivity between content source and consumers. In this case, a content cached at intermediate nodes called content providers may avoid unavailability of the former within a partitioned region. A mobile node that holds the original copy of content is called data source/data provider/server. An Interest for a content is initiated by a consumer node which is forwarded in hop-by-hop manner towards source/provider. Any in-network cache node having same copy of the content may answer requests on behalf of source node. As mentioned earlier, CSCM consists of two phases. In following sections, we explain in detail the working of each phase.

5.2.2 Cache Node Selection

In the CSCM, whenever a consumer node generates a request for content c, it sends an *Interest* packet towards the server. The request is transmitted through neighboring nodes in a multi-hop fashion until it reaches the server. When the server receives the *Interest*, it replies by sending back the content *Data* that also reaches the consumer node in a multi-hop fashion. The consumer's requests can be propagated through some standard controlled flooding protocols [111] and in response the content *Data* is sent back to the consumer by using a single path that is discovered during the request phase.

According to the CCN mechanism, each node is equipped with Cache Storage (CS) that is used to cache incoming content, a Pending Interest Table (PIT) as used to keep track of the of unsatisfied forwarded *Interests* and Forwarding Information Base (FIB) is used as a routing table based on content names. If an *Interest* packet arrives at a node that does not have the content in its cache and a cache *miss* occurs then the request is stored in the PIT table and the *Interest* is forwarded to the neighboring nodes according to the FIB table entries. If the requested content is present in the local cache and a cache *hit* occurs then the content *Data* is sent back to the consumer in a multi-hop fashion.

In CSCM, cache node selection on content delivery path is analogous to cache selection in MACS [99]. We use the same basic methodology defined in MACS [99] (summarized in Table.5.1). However, we introduce new attributes for better

TABLE 5.1: *Interest/Data* received at CR r

ALGORITHM-5: Receiving *Data* Packet

```
1.  IF (Content c arrives) at router $r$ then
2.    Calculate $MCD_{Score}(c)$
3.      IF ($MCD_{Score}(c) > \mu$) then
4.        IF (Cache Space is Full) then
5.          Evict the content
6.          Cache(Content);
7.        EndIF
8.      EndIF
9.    Forward(Content);
10. EndIF
```

Receiving *Interest* Packet

```
1. IF (Intrest $I_c$ for content c arrives) at router $r$ then
2.   IF ($c$ is in local CS) then
3.     Forward(Content $c$);
4.   ElseIF ($I_c$ is not in PIT)
5.     PIT $\leftarrow$ PIT $\cup$ $I_c$
6.     Send($I_c$ to neighboring nodes);
7.   EndIF
8. EndIF
```

adaptability in mobile environments. Here, every node independently calculates its feasibility to store an incoming content for maximizing overall network performance. On receiving a content, an intermediate node first calculates its composite cache's decision score ($MCD_{Score}(c)$). If the composite decision score is greater than a given threshold value (as defined in following Equation. (5.1)), node stores the content in its cache; otherwise, it forwards content to downstream node. Moreover, if the cache is already full, freshness replacement policy is used as discussed in Section. 3.4.7.

$$MCD_{Score}(c) = \begin{cases} 1 & if \quad MCD_{Score}(c) \geq \mu \\ 0 & Otherwise \end{cases} \tag{5.1}$$

Detail discussion on how this threshold value is selected can be described in Section. 3.4.4. Details on calculating composite cache's decision score ($MCD_{Score}(c)$) considering the mobile environment are described in next subsections.

5.2.2.1 Cache Placement Policy

The prime philosophy behind proposed CSCM caching scheme is to store contents on few optimal nodes, instead of caching on each intermediate relay. This helps in saving network resources by minimizing the presence of duplicated copies of a content within the network. The pseudo-code of cache placement is defined in Table. 5.2.

Table 5.2: Pseudocode of Cache Placement

```
PROGRAM CachePlacement:
    Read Packet;
    IF (Interest Packet)
        IF (Content in Cache)
            Forward Content Packet to Downstream Router;
        ElseIF (Interest Packet is not in PIT Table)
            Add Interest in PIT Table;
            Send Interest Packet to Neighboring Nodes ;
        ENDIF;
    Else
        At Current Router Calculate $MCD_{Score}(c)$;
            IF ($MCD_{Score}(c) > \mu$)
                    IF (Cache Space is Full)
                        Remove the Content from Cache;
                        Save the Content in Cache;
                    ENDIF;
            ENDIF;
                Forward Content Packet to Downstream Router;
    ENDIF;
```

In this regard, composite cache's decision score ($MCD_{Score}(c)$) is used to determine suitability of a node for caching contents within the network. We calculate

$MCD_{Score}(c)$ against every incoming content at an intermediate node r_i, as defined in Equation (5.2). For $MCD_{Score}(c)$ calculation, following metrics are considered to cope with challenges of mobile environments: (1) node centrality ($\Theta_{(r_i)}$), (2) available cache space ($\partial_{(r_i)}$), and (3) energy level ($\sigma_{(r_i)}$).

The individual score of each parameter is combined in the composite decision score for maximizing content storing feasibility of a node.

$$MCD_{Score}(c) = w_1 \times \Theta_{(r_i)} + w_2 \times \partial_{(r_i)} + w_3 \times \sigma_{(r_i)} \tag{5.2}$$

The relevant importance of each parameter is determined through weights w, which are used to maximize the content cache decision of a given node. The sum of all weights w_i for different parameters is one as presented in Equation. (5.3).

$$\sum_{i=1}^{p} w_i = 1 \tag{5.3}$$

Each attribute score is a real number within range [0, 1]. A high value of any given attribute score contributes in increasing feasibility of a content router/node r_i for storing an incoming content in its cache. In following, we present more details on calculation of each attribute.

Node Centrality: As per definition of graph theory, the relative important measure of a vertex inside a graph is termed as centrality [112]. This theory has been widely used in research community to identify most influential person(s) in social networking, and also for identifying crucial nodes within computer networks etc. It is one of the most common metrics used for finding important or central nodes within any given network for dynamic bandwidth allocation in vehicular social networks [113], and routing [112] etc. Here, in CSCM caching contents at nodes with high centrality rankings within the network may increase accessibility of a cached content resulting in large cache hits with limited redundancy.

Node centrality is defined according to the Eigenvector Centrality (EC) that implies that " A node is important if it is linked to other important nodes ". Therefore, a node has a high value of EC either if it is connected to many other nodes, or if it is connected to neighbors which have a high EC. The EC of a node is defined as:

$$\Theta_{(i)} = \frac{1}{\lambda} \sum_{j \in M} a_{ij} \Theta_{(j)} \tag{5.4}$$

Where λ is a constant, M is the number of nodes in the network and a_{ij} is the adjacency matrix, where a_{ij} is 1 if two nodes are connected and 0 if not. In our scenario, every node calculates its initial score based on number of current connections (i.e. (no. of connections)/M). This is called socialness of a node within the network. These centrality values are periodically shared among one-hop neighbors to find Eigenvector Centrality as in Equation. (5.4). Therefore, according to Equation.(5.4), if $I's$ neighbors are highly social nodes, then I also have high socialness based on Eigenvector Centrality. In this case, updated centrality values are periodically broadcast among one-hop neighbors to incorporate with topology changes within the network due to mobility.

Available Cache Space: Available storage space at a node is an important parameter to consider when caching decisions are made. This is because forwarding nodes are mobile devices with limited available space [67] [66], [15]. Therefore, it is important to effectively utilize these stringent resources in terms of space for improving overall network performance.

Available cache space $\partial_{(r_i)}$ is estimated at node r_i, as shown in Equation. (5.5). High value of $\partial_{(r_i)}$ at a node represents that the latter has enough free buffer space to be selected as candidate node for caching incoming contents; while nodes with limited free storage space are less suitable, and should be selective in taking caching decisions. Intuitively, the caching probability of a node is inversely proportional to this cache occupancy level.

$$\partial_{(r_i)} = \frac{CacheSize_{(r_i)} - CacheSize_{(r_i)used}}{CacheSize_{(r_i)}} \tag{5.5}$$

Here, $CacheSize_{(r_i)used}$ represents the used cache space on r_i and $CacheSize(r_i)$ is the total cache size at r_i.

Battery Energy Level: Remaining energy is an important factor in resource-constrained networks (i.e. MANET) [64]. Energy is an important resource, as it helps mobile nodes to remain part of the network for long duration before requiring a recharge. Remaining energy level is estimated at a node to determine the amount of time this node can contribute in functioning of the network. Nodes with sufficient energy levels can actively participate in caching operations. On the other hand, nodes with low energy levels may shortly enter in sleep mode to save energy. Therefore, nodes with low energy level are dynamically restrained to participate in caching operations in CSCM. Hence, in CSCM caching probability is directly proportional to residual energy level. Node r_i first determines its consumed energy defined as φ to calculate its remaining energy in Equation. (5.6). We assume that all nodes have equal energy at start of the network. On every operation such as transmitting, receiving, and caching etc. consumes one unit of energy. Therefore, we increment the counter on every operation at a node to determine the consumed energy.

Next, we can calculate remaining energy level $\sigma_{(r_i)}$ at node r_i as follows:

$$\sigma_{(r_i)} = \left\{ \lambda e^{-\lambda(\varphi)} \right. \tag{5.6}$$

5.2.3 Popularity and Freshness-driven Mobility Adaptive Cache Relocation

CSCM is an adaption of existing cache schemes in mobile environments. Traditional caching approaches face performance degradation when applied to challenging environment such as mobile hosts. CSCM is a general framework which might be accompanied by existing approaches to make them suitable for mobile networks.

In CSCM, mobility of a node with cached contents (referred here as (Cache Node (CN)) is classified into two categories: 1) mobility within the region and 2) mobility across the region. Mobility within the region does not effect availability of contents. On the other hand, mobility of CN across the region may result in unavailability of data for future requests.

Therefore, in CSCM when CN is near to move, it selects the next most appropriate candidate node to serve as new CN for this region. After selection, selected cached contents are forwarded to new candidate CN. Now, old CN can move to a new region without effecting content availability within the region. Next, we explain in detail the process of candidate node selection, and content selection for relocation.

5.2.3.1 Candidate Cache Node Selection

For candidate cache node selection, CN regularly check its current status. If CN is about to move, it selects a new CN to relocate cached contents according to $Algorithm - 6$ define in Table. 5.3.

To explain working of the algorithm, we take the scenario shown in Figure. 5.1. Figure. 5.1 shows a working example of how candidate cache node selection takes place. Router/node A is the *Cache Node* which checks its current status and identifies that it is *Near to Move* (Table. 5.3 line(1)). It broadcasts *Announcement* message to its one-hop neighbor routers/nodes (Table. 5.3 line(2)). *Announcement* message contains a selected list of stored contents (content selection is discussed in next sections) on$A$. Every one-hop routers calculate the MCD_{score} value and compare it with threshold. If MCD_{score} is greater than the threshold value and the node's $CurrentStatus$ is not Near to Move, then it sends the Interest packet (Table. 5.3 line(7-8)) to A(as shown on node B, D, and F). At node A, after receiving the first Interest packet from its neighbors (Table. 5.3 line (4)) (as node F), node A forwards *Data* packets (Table 5.3 line (5)) to new CN i.e. node F on Figure. 5.1. Receiving node (i.e. node F), receives *Data* packets and store in its cache(Table. 5.3 line (9-10)). In case, the value is lower than the current

Table 5.3: Candidate-Cache Node Selection at CR 'r'

ALGORITHM-6

At Content Provider

```
1.    IF ($Cache\_Node_{(CurrentStatus)} = Near\ to\ Move$) Then
2.        $Announcement(Available\, list\, of\, contents)$ \\ Send to its one-hop neighbors
3.        Wait for Interest
4.        Receive $Interest(Content)$ \\ from its one-hop interested neighbors
5.        Send $Data(Content)$
6.    EndIF
```

At Neighboring Node

```
7.    IF($MCD_{score} \geq \mu \,\&\&\, CurrentStatus! = Near\, to\, Move$) then
8.        Send $Interest(Content)$
9.        Receive $Data(Content)$
10.       Save $Data(Content)$
11.   Else
12.       Broadcast $Announcement(Available\, list\, of\, contents)$
13.   EndIF
```

threshold, the node further broadcasts the announcement to its one-hop neighbors (Table. 5.3 line (12)). The pseudo-code of candidate cache node selection is defined in Table. 5.4.

The operation flow of the candidate cache node selection is shown in Figure. 5.2.

5.2.3.2 Data Selection for Relocation

When a node is *Near to Move*, relocating all the cached contents may cause extra overhead over the network. We suggest that popular and fresh data is a good choice for relocation. On the current node, content popularity is computed by

TABLE 5.4: Pseudocode of Candidate Cache Node Selection

```
PROGRAM CandidateSelection:
    Read Cache Node Current Status;
    IF (It is Near to Move)
        Announce its Available Content List to its Neighbors;
        Neighboring Nodes Calculate its $MCD_{Score}(c)$;
        IF ($MCD_{Score}(c) > \mu$)
            Send Interest Packet of Content to Content Provider;
            Receive Data Packets from Content Provider;
            Save Data Packets in its Cache;
        Else;
            Broadcast Announcement its Available Content List;
        ENDIF;
    ENDIF;
```

considering both the frequency and freshness of the content. $F(x)$ is a weighting function that is defined as:

$$F(x) = (\frac{1}{2})^{\theta x} \tag{5.7}$$

Where θ is the control parameter and its range is from 0 to 1. This control parameter θ allows a trade-off between recency and frequency. As $\theta = 0$, frequency has a greater influence on the content popularity than recency and when $\theta = 1$, recency has a greater influence on the content popularity than frequency. To achieve a good trade-off between recency and frequency, we set θ value as e^{-4} [114]. Popularity of content i is defined as in Equation. (5.8) that is based on past n requests to this content [114]. Let x be defined as:

$$x = \sum_{j=1}^{n} (t_{base} - t_j) \tag{5.8}$$

Where t_{base} is the current time and t_j is denoted as the past arrival time of the request for content i. Let $\{t_1, t_2, ..., t_j\}$ are the system time which is assumed that these are integer values such that $t_1 < t_2, ..., < t_k \leq t_{base}$.

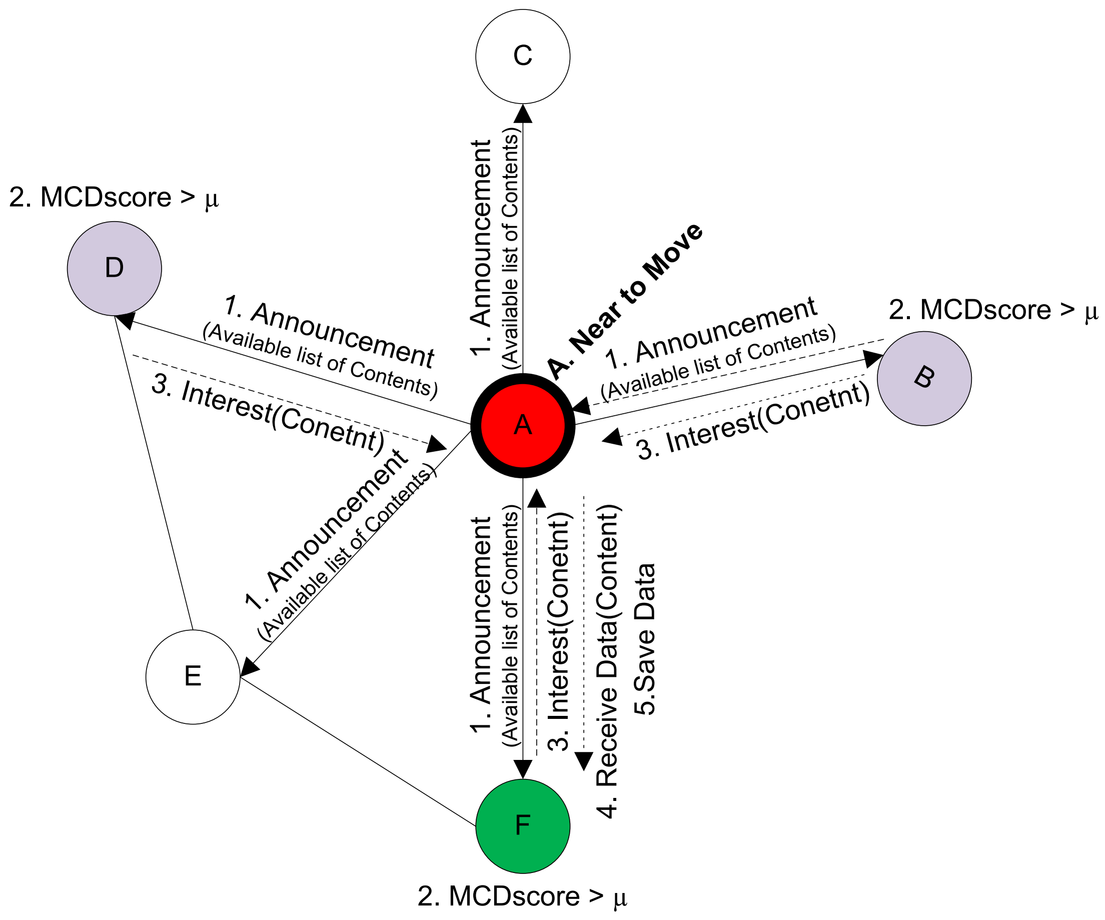


FIGURE 5.1: An example of candidate cache node selection

A content is selected for relocation, if its weighting function defined in Equation. (5.8) is greater than given threshold value.

5.2.4 Complexity Analysis of CSCM

Time and space complexity are other two important considerations to be made for deployment in real world situations. In CSCM, time complexity of calculating an individual score of multiple parameters is not more than $\mathcal{O}(n)$. As we use additive property, time complexity of scores estimation is equal to the highest time complexity of calculating any individual score. The parameters we use in this thesis such as node centrality, cache space and energy level etc., scores have constant time complexity. Hence, time complexity of scores estimation in CSCM of constant time i.e. $\mathcal{O}(1)$. Time complexity of weight adaptation in CSCM is $\mathcal{O}(n)$

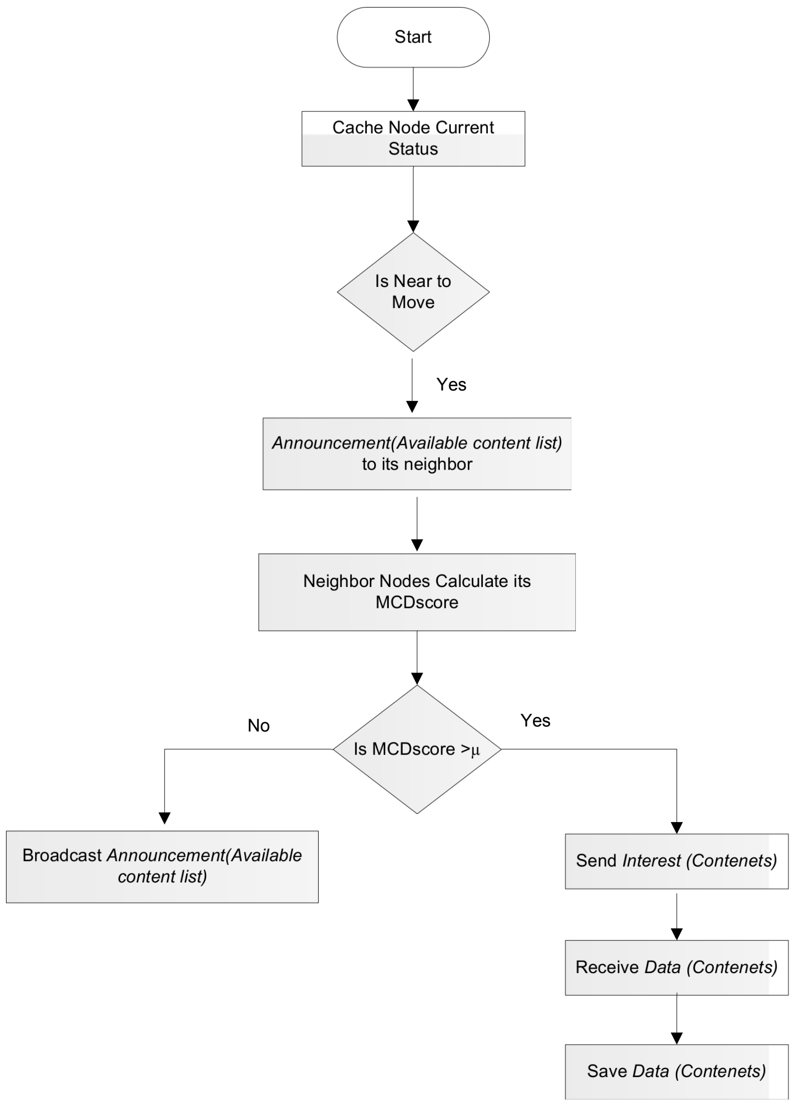

Start
Cache Node Current Status
Is Near to Move
Yes
Announcement(Available content list) to its neighbor
Neighbor Nodes Calculate its MCDscore
No
Is MCDscore >µ
Yes
Broadcast *Announcement(Available content list)*
Send *Interest (Contenets)*
Receive *Data (Contenets)*
Save *Data (Contenets)*

FIGURE 5.2: A flow diagram of Candidate Cache Node Selection

for n attributes. Hence, we can safely say that the time complexity of CSCM is $\mathcal{O}(n)$.

Coming to space complexity of our proposed scheme, evaluations are made on most recent available information from interest/data packets such as energy level and cache space. There is no need of extra space at the node to keep record of information. Hence, worst space complexity of CSCM is $\mathcal{O}(n)$ where n is the total number of attributes.

5.3 Simulation and Performance Evaluation

We present simulation results in this section, and demonstrate the performance of CSCM by evaluating it against a number of performance metrics such as [67], [62]:

- **Network Traffic** : Network traffic is the sum of *Interest* packets and *Data* packets that are used to evaluate network load.
- **Retrieval Time** : Content retrieval time is defined as the time duration when an *Interest* packet is sent, and when the corresponding *Data* packet is received. It denotes the response latency that is used to evaluate reduction in response time because of using any in-network caching strategy.
- **Average Hit Rate** : Average hit rate is the ratio of number of requests, answered by source node to total number of requests generated within the network.
- **Relocation Frequency** : Relocation frequency is the number of times contents are relocated from one node to another node due to mobility.

Moreover, we compare the performance of CSCM with MACS scheme that is Multi-Attribute Caching Strategy (MACS) for CCN.

5.3.1 Simulation Setup

To evaluate performance of proposed CSCM caching strategy, we perform simulations in open-source ndnSIM simulator[**?**] that implements NDN protocol stack for NS-3 network simulator.

In MANET environment, 60 nodes are randomly distributed in 1000m x 1000m area. We use random waypoint model for node mobility. Initially, nodes are stationary and after some time randomly move to change position at the speed of $0 \sim 30m/s$. Among nodes one data provider is selected randomly, and remaining

TABLE 5.5: Simulation parameters

Attribute	**Value**
Area size	1000m x 1000m
Number of different content	100
Total number of mobile nodes	60
Cache size (number of contents)	10
Mobility model	Random waypoint
Maximum residence time	8s
Simulation time	200s
Simulation runs	5
μ	0.6
Maximum rate of node movement	$0 \sim 30m/s$
MAC type	Ad-hocWifiMac

node are taken as consumers. Consumer nodes randomly generate *Interest* packets following Poisson process, and BestRoute forwarding strategy is used to forward *Interest* packet towards provider node. The requested content use zipf distribution. Zipf(α) is implemented as content popularity distribution that is simply tuned by a single parameter α. Content request q is modeled as Poisson processes [115] and the average interest packet rate is taken as 20 packets/s.

Simulation results are taken as average with with the 95% confidence intervals by running each scenario five times with different seed values. Defining an appropriate value for threshold is important to obtain maximum performance. However, it varies based on network conditions of the underlying scenario. Hence, we suggest choosing threshold taking into account available resources and performance trade-offs. For this purpose, we perform a number of simulation experiments to define a suitable value for mobile ad-hoc environments. The threshold value is set to $\mu = 0.6$ after performing extensive simulations. The Simulation parameters are shown in Table. 5.5 [67].

5.3.1.1 Network Traffic

Figure. 5.3 shows network traffic comparison of proposed CSCM and MACS scheme against simulation time.

At start of simulation, network traffic generated by both schemes is high. It is because, at start, there is no data cached at intermediate nodes and all the Interests are forwarded to source, which cause high network traffic. With the passage of simulation time, data is cached at intermediate nodes due to which some requests are answered through intermediate nodes. This decrease the overall network traffic at the original source. Both the schemes use same mechanism to store contents on intermediate nodes, thereby, resulting in same behavior over time. However, at 50 sec, nodes that have cached contents move around other parts of the network. MACS has no in-built feature for handling cached contents in case of mobility. However, in case of CSCM, cached contents are relocated on neighbor nodes. Hence, these are delivered to requesting nodes from cache instead of sending all the Interests to the original content provider. Thereby, overall network traffic is significantly reduced in case of CSCM.

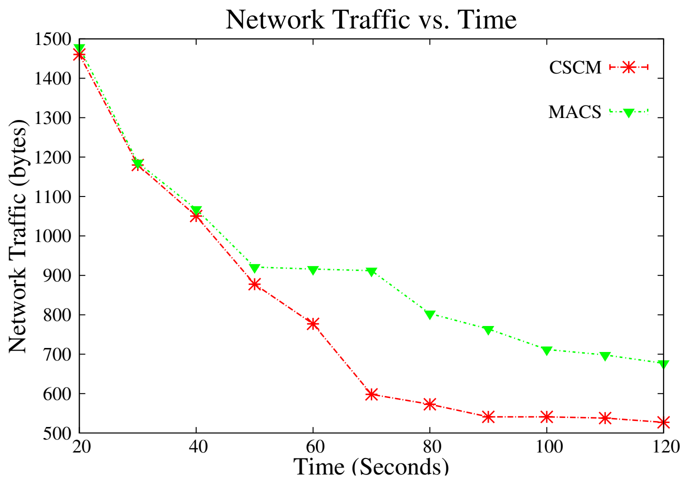


FIGURE 5.3: Comparison of network traffic

5.3.1.2 Average Hit Rate

Average hit rate on server is used to determine cache efficiency of in-network caching strategies. Figure. 5.4 shows results of average hit rate comparison between CSCM, and MACS. Average hit rate is plotted against ratio of requesting nodes. When requesting nodes are less, average hit rate at the content provider is also low. With increase in number of content requesting nodes, average hit rate is also increasing. It is because, increasing number of requesting nodes generate large number of Interests that are forwarded to source causing high hit ratio at the latter. The average hit ratio of CSCM is low as compared to MACS because most of the Interests are answered by neighboring nodes in CSCM in case of mobility of a cache node. In other words, Interests from that region are satisfied with relocated contents instead of sending them to content source as compare to MACS. As a result, average hit ratio at the source is decreased. However, MACS strategy does not consider content relocation, hence, cached contents are lost from the region due to network partitioning. Therefore, requests were forwarded to the original source causing high hit rate at the source.

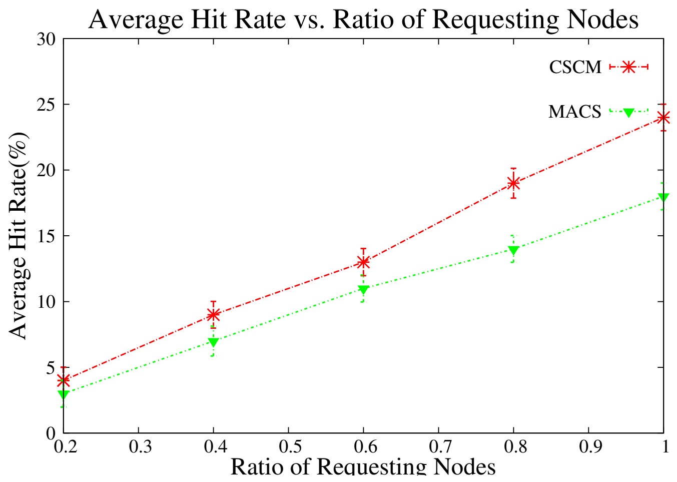


FIGURE 5.4: Comparison of average hit rate

5.3.1.3 Retrieval Time

Figure. 5.5 depicts the results for average retrieval time comparison of CSCM and MACS against simulation time. At start of simulation, the retrieval time of both schemes is high. This is high because, at start, there is no data cached at intermediate nodes, and Interests are forwarded to the source that cause high data retrieval time. As simulation time increases, contents are replicated on intermediate nodes. Some of the Interests are served by local content store of intermediate caching nodes instead of sending all the Interests to the content provider. This decreases the overall content retrieval time. The cache used in CSCM strategy relocate contents to neighboring nodes, so retrieval time of CSCM is low as compared to MACS. With the passage of time, majority of the nodes serve as cache nodes and their mobility has no severe effect on availability of the content, as it is already available in the region. Therefore relocation of contents show low performance improvements for retrieval time.

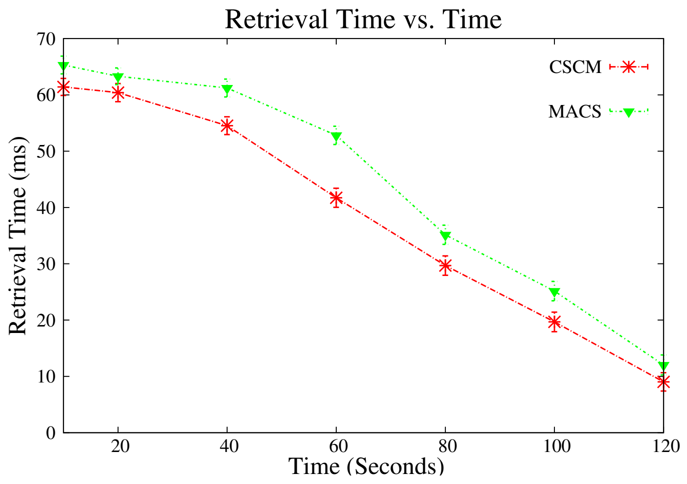


FIGURE 5.5: Comparison of retrieval time

5.3.1.4 Relocation Frequency

Transmission range is directly proportional to connectivity of nodes within the network. With small transmission ranges, a slight movement of nodes may cause significant changes in network topology due to differences in nodes connectivity. Movement/disconnection of a node from its current neighbors towards a new location requires relocation of data. Therefore, this event is significant at smaller transmission ranges and vice versa. We compare the proposed techniques against varying transmission ranges to identify the behavior against varying patterns of disconnections within the network.

Figure. 5.6 shows results for relocation frequency of replication. Relocation frequency is plotted against different transmission ranges. The graph for relocation frequency depicts how many times data are relocated from an existing cache node to newly selected node, when transmission range between nodes is varying. Simulation is run for 200 seconds and nodes are moving with speed of 15 m/sec. With the increase in transmission range between two nodes, the relocation frequency is found decreasing. This is because, due to large transmission range nodes can freely move without changing connectivity. On the other hand, if transmission range between two nodes is limited, and provider node moves to a far away region. This may result in disconnectivity between service provider and consumer resulting an increase in number of required relocations. However, if transmission range increases between the nodes, multi-hop communication may cover larger areas to provide connectivity of service. As a result, service provider moves but stays connected with consumers through multi-hop communication. Therefore, due to improved connectivity within the networks number of relocations are reduced.

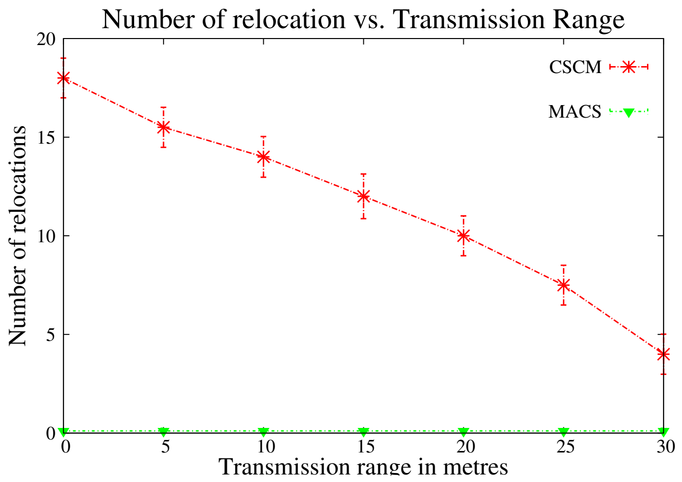


FIGURE 5.6: Comparison of relocation frequency varying transmission ranges between node pairs

5.4 Conclusion

In this chapter, a caching strategy for CCN-MANET is proposed, called CSCM, that improves caching efficiency and to reduce overhead. The main purpose of the proposed caching scheme is to cache the contents in the core nodes and to reduce the redundancy of data in the network. CSCM based on two phases, first, to dynamically adapt the caching decision of each content by considering three main parameters: the battery energy level, the buffer space of the node, and the node centrality. Second, the caching situation depends on a node's current status, popularity and freshness of the data that affects the caching decision in order to relocate the replica if the old cached node moves to another location.

We have evaluated our proposed framework, CSCM with MACS caching scheme. Basic MACS has no built in feature to handle cached contents in case of mobility. Whereas, in CSCM, the movement of small number of cached content nodes relocated the content to neighbor nodes for further interests. Simulation results proved that the CSCM strategy can reduce content retrieval delays and the network traffic load. It also considered the most stable and centralized node to replicate the data. The way we compute the content caching decision in CSCM is very versatile because it can be easily modified by adding other parameters.

Chapter 6

Conclusion and Future Work

CCN is a promising technology for data dissemination that can be used in wired and wireless ad-hoc networks. The default caching strategy of CCN is to store the content at each content-router (CR) along the downloading path. While this helps to increase content availability and quality-of-experience (QoE) by reducing delay and the server load. Therefore, the ubiquitous in-network caching strategy is not resource-efficient as it increases the cache redundancy unnecessarily. A number of cache management schemes have been proposed to address the issue of resource utilization in CCN, and most of these schemes consider only one attribute of network while taking the content placement decision. The result to use these single parameter techniques is sub-optimal network performance that may be useful for one scenario and may not be as effective for another. The resource utilization can be improved by considering multiple attributes simultaneously for cache management that can give better network performance even with dynamic environments.

In the first part of this thesis (Chapter 3), we proposed a Multi-Attribute Caching Strategy (MACS) for Content-Centric Networks that determines suitable caching location along the content delivery paths while taking into account multi-dimensional characteristics. MACS considers different aspects of a network such as, global topology, node connectivity, and content-based information. MACS makes independent caching decisions at each content router and selects a suitable caching location along the content delivery path. Moreover, MACS calculates relative importance of each dimension through a dynamic weight adaption mechanism that is based on error correction learning. These attributes with associated weights are then employed to determine the probability to store content on returning path. If the probability is greater than a defined threshold value, content is stored, otherwise, it is simply forwarded to next downstream router. The parameters which we are considering in this thesis include content popularity, degree of node, hop-count/distance and cache storage space.

We show with simulations that MACS dominates among other schemes in efficient resources utilization. MACS ability to carefully store a content, where it is most popular helps in delivering satisfactory performance in resource-stringent environments. However, in MACS the tendency to effectively utilize the available resources comes with a slight compromise on hop-reduction ratio.

The advantages of MACS can be summarized as follows:

- MACS is the caching mechanism that determines the different aspects of content's and network topology and utilizes them to calculate the cache utility value.
- To determine, dynamically relative importance of each attribute, MACS uses the error correction learning [19] technique of neural networks.
- To maintain freshness of cached content and to keep fresh contents in cache, MACS defines a method that dynamically update freshness of a content.
- A new approach for cache replacement is also introduced that is based on freshness value of a cache content.

In the second part of the thesis (Chapter 5), we studied in-network caching in CCN-based mobile ad-hoc network, and proposed a distributed caching scheme called Caching Strategy in CCN-MANET (CSCM). CSCM is a complete mobility framework to handle mobility of in-network cache nodes. It is divided into two phase; first, it takes caching decision by considering three main parameters, 1) node's centrality, 2) energy level, and 3) cache space of a node. Secondly, in case of mobility, it selects a new cache node to relocate replications of cached contents, if the old cache node intends to move towards another region.

For the first phase, CSCM is flexible of working with existing in-network cache schemes. For this purpose, we adapted MACS (in-network cache) for mobile environments, which is another significant contribution of this paper. Simulation results show that CSCM strategy can reduce content retrieval delays, and traffic load over the data source. The way we compute content caching decision in CSCM is very scalable, because it can be easily modified to add new parameters.

The work can be extended as future work: 1) by considering more aspects of device and content-based information in contents caching decision and 2) extending the CSCM scheme for opportunistic network scenarios that will collect a history of encountered nodes and will compute statistics about the distribution of contents across the network. The acquired knowledge about the network and content will enable content routers to build up an internal representation of the current state of the content router and the network. This knowledge-based approach will use to improve the decision-making process of content caching and dissemination. Moreover, we will design a cache relocation mathematical modeling in CCN-MANET. This framework will be enhanced to apply to real MANETs environments.

Bibliography

[1] V. N. I. Cisco, "Forecast and methodology, 2015-2020," *White Paper, Cisco*, 2016.

[2] E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks," in *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4. ACM, 2002, pp. 177–190.

[3] A. Wierzbicki, N. Leibowitz, M. Ripeanu, and R. Wozniak, "Cache replacement policies revisited: The case of P2P traffic," in *Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symposium on.* IEEE, 2004, pp. 182–189.

[4] G. Zhang, M. Tang, S. Cheng, G. Zhang, H. Song, J. Cao, and J. Yang, "P2p traffic optimization," *Science China Information Sciences*, vol. 55, no. 7, pp. 1475–1492, 2012.

[5] 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*, p. 4, 2007.

[6] A. A. Barakabitze and T. Xiaoheng, "Caching and data routing in information centric networking (icn): The future internet perspective," *International Journal*, vol. 4, no. 11, 2014.

[7] 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.

[8] 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.

[9] G. Zhang, Y. Li, and T. Lin, “Caching in information centric networking: a survey,” *Computer Networks*, vol. 57, no. 16, pp. 3128–3141, 2013.

[10] 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*. ACM, 2009, pp. 1–12.

[11] 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, no. 4. ACM, 2007, pp. 181–192.

[12] M. Ain, D. Trossen, P. Nikander, S. Tarkoma, K. Visala, K. Rimey, T. Burbridge, J. Rajahalme, J. Tuononen, P. Jokela *et al.*, “D2. 3–architecture definition, component descriptions, and requirements,” *Deliverable, PSIRP 7th FP EU-funded project*, 2009.

[13] B. Ahlgren, M. Dambrosio, C. Dannewitz, A. Eriksson, J. Golic, B. Grönvall, D. Horne, A. Lindgren, O. Mämmelä, M. Marchisio *et al.*, “Second netinf architecture description,” *4WARD EU FP7 Project, Deliverable D-6.2 v2. 0*, 2010.

[14] S. Borst, V. Gupt, and A. Walid, “Distributed caching algorithms for content distribution networks,” in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[15] D. Rossi, G. Rossini *et al.*, “On sizing CCN content stores by exploiting topological information.” in *INFOCOM Workshops*, 2012, pp. 280–285.

[16] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache less for more in information-centric networks," in *NETWORKING 2012*. Springer, 2012, pp. 27–40.

[17] A. Ioannou and S. Weber, "A taxonomy of caching approaches in information-centric network architectures," *Elsevier Journal*, 2013.

[18] 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*. ACM, 2012, pp. 55–60.

[19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[20] "Ndnsim project [eb/ol]," 2012. [Online]. Available: http://ndnsim.net/

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

[22] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Modeling data transfer in content-centric networking," in *Proceedings of the 23rd international teletraffic congress*. International Teletraffic Congress, 2011, pp. 111–118.

[23] L. Muscariello, G. Carofiglio, and M. Gallo, "Bandwidth and storage sharing performance in information centric networking," in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*. ACM, 2011, pp. 26–31.

[24] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie, "Design and evaluation of the optimal cache allocation for content-centric networking," *IEEE Transactions on Computers*, vol. 65, no. 1, pp. 95–107, 2016.

[25] C. Bernardini, T. Silverston, and A. Vasilakos, "Caching strategies for information centric networking: Opportunities and challenges," *arXiv preprint arXiv:1606.07630*, 2016.

[26] S. Arianfar, P. Nikander, and J. Ott, "Packet-level caching for information-centric networking," in *ACM SIGCOMM, ReArch Workshop*, 2010.

[27] A. Kalla and S. K. Sharma, "A constructive review of in-network caching: A core functionality of icn," in *Computing, Communication and Automation (ICCCA), 2016 International Conference on.* IEEE, 2016, pp. 567–574.

[28] I. U. Din, S. Hassan, M. K. Khan, M. Guizani, O. Ghazali, and A. Habbal, "Caching in information-centric networking: Strategies, challenges, and future research directions," *IEEE Communications Surveys & Tutorials*, no. 99, p. 1, 2017.

[29] G. Rossini and D. Rossi, "A dive into the caching performance of content centric networking," in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2012 IEEE 17th International Workshop on.* IEEE, 2012, pp. 105–109.

[30] T.-A. Le, N. D. Thai, P. L. Vo *et al.*, "The performance of caching strategies in content centric networking," in *Information Networking (ICOIN), 2017 International Conference on.* IEEE, 2017, pp. 628–632.

[31] D. Rossi and G. Rossini, "Caching performance of content centric networks under multi-path routing (and more)," *Relatório técnico, Telecom ParisTech*, pp. 1–6, 2011.

[32] Y. Xu, S. Ci, Y. Li, T. Lin, and G. Li, "Design and evaluation of coordinated in-network caching model for content centric networking," *Computer networks*, vol. 110, pp. 266–283, 2016.

[33] C. Li and K. Okamura, "Cluster-based in-networking caching for content-centric networking," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 14, no. 11, p. 1, 2014.

[34] A. Jiang and J. Bruck, "Optimal content placement for en-route web caching," in *Network Computing and Applications, 2003. NCA 2003. Second IEEE International Symposium on.* IEEE, 2003, pp. 9–16.

[35] Z. Li and G. Simon, "Time-shifted TV in content centric networks: The case for cooperative in-network caching," in *Communications (ICC), 2011 IEEE International Conference on.* IEEE, 2011, pp. 1–6.

[36] J. M. Wang, J. Zhang, and B. Bensaou, "Intra-as cooperative caching for content-centric networks," in *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking.* ACM, 2013, pp. 61–66.

[37] M. . M. M. F. . K. J. Wong, Walter & Giraldi, "Content routers: Fetching data on network path," in *Communications (ICC), 2011 IEEE International Conference on.* IEEE, 2011, pp. 1–6.

[38] W. K. Wong, L. Wang, and J. Kangasharju, "Neighborhood search and admission control in cooperative caching networks," in *Global Communications Conference (GLOBECOM), 2012 IEEE.* IEEE, 2012, pp. 2852–2858.

[39] M. Bilal and S.-G. Kang, "A cache management scheme for efficient content eviction and replication in cache networks," *IEEE Access*, vol. 5, pp. 1692–1701, 2017.

[40] S. Eum, K. Nakauchi, M. Murata, Y. Shoji, and N. Nishinaga, "Catt: potential based routing with content caching for ICN," in *Proceedings of the second edition of the ICN workshop on Information-centric networking.* ACM, 2012, pp. 49–54.

[41] Y. Li, T. Lin, H. Tang, and P. Sun, "A chunk caching location and searching scheme in content centric networking," in *Communications (ICC), 2012 IEEE International Conference on.* IEEE, 2012, pp. 2655–2659.

[42] J. Guan, Y. He, Q. Wei, and Z. Neng, "A classification-based wisdom caching scheme for content centric networking," in *Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on.* IEEE, 2016, pp. 822–827.

[43] 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," *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 2010.

[44] 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.

[45] X. Hu and J. Gong, "Opportunistic on-path caching for named data networking," *IEICE Transactions on Communications*, vol. 97, no. 11, pp. 2360–2367, 2014.

[46] 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.

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

[48] H. Xiaoyan and G. Jian, "Opportunistic on-path caching for named data networking," *IEICE Transactions on Communications*, vol. 97, no. 11, pp. 2360–2367, 2014.

[49] N. Abani, G. Farhadi, A. Ito, and M. Gerla, "Popularity-based partial caching for information centric networks," in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2016 Mediterranean.* IEEE, 2016, pp. 1–8.

[50] S. Arianfar, P. Nikander, and J. Ott, "On content-centric router design and implications," in *Proceedings of the Re-Architecting the Internet Workshop.* ACM, 2010, pp. 5–15.

[51] G. P. Mishra and M. Dave, "Cost effective caching in content centric networking," in *India Conference (INDICON), 2015 Annual IEEE.* IEEE, 2015, pp. 1–5.

[52] 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.* IEEE, 2012, pp. 268–273.

[53] ——, "Age-based cooperative caching in information-centric networking," in *Computer Communication and Networks (ICCCN), 2014 23rd International Conference on.* IEEE, 2014, pp. 1–8.

[54] C. Bernardini, T. Silverston, and O. Festor, "Mpc: Popularity-based caching strategy for content centric networks," in *2013 IEEE International Conference on Communications (ICC).* IEEE, 2013, pp. 3619–3623.

[55] J. Garcia-Reinoso, I. Vidal, D. Diez, D. Corujo, and R. L. Aguiar, "Analysis and enhancements to probabilistic caching in content-centric networking," *The Computer Journal*, vol. 58, no. 8, pp. 1842–1856, 2015.

[56] A. S. Gill, L. D'Acunto, K. Trichias, and R. van Brandenburg, "Bidcache: Auction-based in-network caching in icn," in *Globecom Workshops (GC Wkshps), 2016 IEEE.* IEEE, 2016, pp. 1–6.

[57] K. Hizbullah, U. A. Noor, u. D. Ikram, Insafullah, and I. Jawaid, "Leafpopdown: Leaf popular down caching strategy for information-centric networking," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 9, no. 2, pp. 148–151, 2018.

[58] W. Han, G. Wang, and P. Ying, "Energy-efficiency aware cooperative caching strategy for content-centric networks," in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage.* Springer, 2017, pp. 389–398.

[59] G. Zhang, B. Tang, X. Wang, and Y. Wu, "An optimal cache placement strategy based on content popularity in content centric network," *Journal of Information & Computational Science*, pp. 2759–2769, 2014.

[60] S. Tarnoi, K. Suksomboon, W. Kumwilaisak, and Y. Ji, "Performance of probabilistic caching and cache replacement policies for content-centric networks," in *Local Computer Networks (LCN), 2014 IEEE 39th Conference on.* IEEE, 2014, pp. 99–106.

[61] T. Kawano, M. Shimamura, and H. Koga, "A selective caching scheme based on request history in content-centric networks," in *Proceedings of the 2013 workshop on Student workhop.* ACM, 2013, pp. 47–48.

[62] Y. Zeng and X. Hong, "A caching strategy in mobile ad hoc named data network," in *Communications and Networking in China (CHINACOM), 2011 6th International ICST Conference on.* IEEE, 2011, pp. 805–809.

[63] H. Jin, D. Xu, C. Zhao, and D. Liang, "Information-centric mobile caching network frameworks and caching optimization: a survey," *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, pp. 33–45, 2017.

[64] R. A. Rehman and K. Byung-Seo, "Location-aware forwarding and caching in ccn-based mobile ad hoc networks," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 5, pp. 1388–1391, 2016.

[65] S.-B. Lee, S. H. Y. Wong, K.-W. Lee, and S. Lu, "Content management in a mobile ad hoc network: beyond opportunistic strategy," *International Journal of Communication Networks and Distributed Systems*, vol. 10, no. 2, pp. 123–145, 2013.

[66] M. A. Hail, M. Amadeo, A. Molinaro, and S. Fischer, "Caching in named data networking for the wireless internet of things," in *Recent Advances in Internet of Things (RIoT), 2015 International Conference on.* IEEE, 2015, pp. 1–6.

[67] L. Zhang, J. Zhao, and Z. Shi, "Lf: A caching strategy for named data mobile ad hoc networks," in *Proceedings of the 4th International Conference on Computer Engineering and Networks.* Springer, 2015, pp. 279–290.

[68] Y. Liu, D. Zhu, and W. Ma, "A novel cooperative caching scheme for content centric mobile ad hoc networks," in *Computers and Communication (ISCC), 2016 IEEE Symposium on.* IEEE, 2016, pp. 824–829.

[69] L. Zhou, T. Zhang, X. Xu, Z. Zeng, and Y. Liu, "Broadcasting based neighborhood cooperative caching for content centric ad hoc networks," in *2015 IEEE/CIC International Conference on Communications in China (ICCC).* IEEE, 2015, pp. 1–5.

[70] K. Chitra and M. Manoj, "Performance improvement in disruption tolerant network with co-operative caching," *IJIRST–International Journal for Innovative Research in Science & Technology*, vol. 1, no. 11, 2015.

[71] A. Rao, P. Kumar, and N. Chauhan, "Cooperative caching strategies for manets and imanets," *Department of Computer Science and Engineering, National Institiute of Technology, Hamirpur, India.*

[72] Y. Wang, J. Wu, and M. Xiao, "Hierarchical cooperative caching in mobile opportunistic social networks," in *Global Communications Conference (GLOBECOM), 2014 IEEE.* IEEE, 2014, pp. 411–416.

[73] P. Costa, C. Mascolo, M. Musolesi, and G. Picco, "Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 26, pp. 748–760, 2008.

[74] M. Conti and M. Kumar, "Opportunities in opportunistic computing," *Computer*, vol. 43, no. 1, 2010.

[75] F. Xia, L. Liu, J. Li, J. Ma, and A. V. Vasilakos, "Socially aware networking: A survey," *IEEE Systems Journal*, vol. 9, no. 3, pp. 904–921, 2015.

[76] C. Boldrini, M. Conti, and A. Passarella, "Contentplace: social-aware data dissemination in opportunistic networks," in *Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems.* ACM, 2008, pp. 203–210.

[77] E. Yoneki, P. Hui, S. Chan, and J. Crowcroft, "A socio-aware overlay for publish/subscribe communication in delay tolerant networks," in *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems.* ACM, 2007, pp. 225–234.

[78] C. Bernardini, T. Silverston, and O. Festor, "Socially-aware caching strategy for content centric networking," in *Networking Conference, 2014 IFIP.* IEEE, 2014, pp. 1–9.

[79] G. Moualla, P. A. Frangoudis, Y. Hadjadj-Aoul, and S. Ait-Chellouche, "A bloom-filter-based socially aware scheme for content replication in mobile ad hoc networks," in *Consumer Communications & Networking Conference (CCNC), 2016 13th IEEE Annual.* IEEE, 2016, pp. 359–365.

[80] C. Yanqing, W. Muqing, Z. Min, and W. Kaili, "Socially-aware noderank-based caching strategy for content-centric networking," in *Wireless Communication Systems (ISWCS), 2016 International Symposium on.* IEEE, 2016, pp. 297–303.

[81] A. N. Langville and C. D. Meyer, "A survey of eigenvector methods for web information retrieval," *SIAM review*, vol. 47, no. 1, pp. 135–161, 2005.

[82] T. Amjad, Y. Ding, J. Xu, C. Zhang, A. Daud, J. Tang, and M. Song, "Standing on the shoulders of giants," *Journal of Informetrics*, vol. 11, no. 1, pp. 307–323, 2017.

[83] J. A. Khan and Y. Ghamri-Doudane, "Saving: socially aware vehicular information-centric networking," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 100–107, 2016.

[84] J. A. Khan, Y. Ghamri-Doudane, and D. Botvich, "Inforank: Information-centric autonomous identification of popular smart vehicles," in *Vehicular Technology Conference (VTC Fall), 2015 IEEE 82nd.* IEEE, 2015, pp. 1–6.

[85] J. A. Khan and Y. Ghamri-Doudane, "Car rank: An information-centric identification of important smart vehicles for urban sensing," in *Network*

Computing and Applications (NCA), 2015 IEEE 14th International Symposium on. IEEE, 2015, pp. 184–191.

[86] J. A. Khan, Y. Ghamri-Doudane, and D. Botvich, “Grank-an information-centric autonomous and distributed ranking of popular smart vehicles,” in *Global Communications Conference (GLOBECOM), 2015 IEEE.* IEEE, 2015, pp. 1–7.

[87] N. Chand, R. C. Joshi, and M. Misra, “Cooperative caching strategy in mobile ad hoc networks based on clusters,” *Wireless Personal Communications*, vol. 43, no. 1, pp. 41–63, 2007.

[88] T. Le, Y. Lu, and M. Gerla, “Social caching and content retrieval in disruption tolerant networks (dtns),” in *Computing, Networking and Communications (ICNC), 2015 International Conference on.* IEEE, 2015, pp. 905–910.

[89] D. P. Dsouza and M. Sujatha, “Survey on ccl based co-operative caching in disruption tolerance networks,” *International Journal of. Emerging Research in Management & Technology*, vol. 5, no. 5, 2016.

[90] T. Wei, L. Chang, B. Yu, and J. Pan, “Mpcs: A mobility/popularity-based caching strategy for information-centric networks,” in *Global Communications Conference (GLOBECOM), 2014 IEEE.* IEEE, 2014, pp. 4629–4634.

[91] T. Vimal and S. S. A. Mary, “Efficient caching and mobility adaptive popular data access in mobile ad hoc networks,” *Indian Journal of Science and Technology*, vol. 9, no. 20, 2016.

[92] T. Sato and S. Goto, “Selective caching with hop counts in information centric networking,” *Proceedings of the Asia-Pacific Advanced Network*, vol. 40, pp. 103–109, 2015.

[93] M. D. Ong, M. Chen, T. Taleb, X. Wang, and V. Leung, “Fgpc: fine-grained popularity-based caching design for content centric networking,” in *Proceedings of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems.* ACM, 2014, pp. 295–302.

[94] C. Bernardini, T. Silverston, and O. Festor, “A comparison of caching strategies for content centric networking,” in *Global Communications Conference (GLOBECOM), 2015 IEEE.* IEEE, 2015, pp. 1–6.

[95] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie, “Optimal cache allocation for content-centric networking.” in *ICNP*, 2013, pp. 1–10.

[96] H. Wang, G. Min, J. Hu, H. Yin, and W. Miao, “Caching of content-centric networking under bursty content requests,” in *2014 IEEE Wireless Communications and Networking Conference (WCNC).* IEEE, 2014, pp. 2522–2527.

[97] I. Kim, Yusung & Yeom, “Performance analysis of in-network caching for content-centric networking,” *Computer Networks*, vol. 57, no. 13, pp. 2465–2482, 2013.

[98] 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.* ACM, 2012, pp. 15–26.

[99] R. N. B. R. . A. Q. Sheneela Naz, “Multi-attribute caching: Towards efficient cache management in content-centric networks,” in *13th IEEE Annual Consumer Communications & Networking Conference (CCNC).* IEEE, 2016, pp. 637–640.

[100] O. Macchi and E. Eweda, “Second-order convergence analysis of stochastic adaptive linear filtering,” *IEEE Transactions on Automatic Control*, vol. 28, no. 1, pp. 76–85, 1983.

[101] S. Haykin and N. Network, “A comprehensive foundation,” *Neural networks*, vol. 2, no. 2004, pp. 41–50, 2004.

[102] A. P. Engelbrecht, *Computational intelligence: an introduction.* John Wiley & Sons, 2007.

[103] J. Quevedo, D. Corujo, and R. Aguiar, "Consumer driven information freshness approach for content centric networking," in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on.* IEEE, 2014, pp. 482–487.

[104] "Geant topology," *http://archive.geant.net/upload/pdf/Topology-Oct-2004.pdf.*

[105] S. Hassan, I. U. Din, A. Habbal, and N. H. Zakaria, "A popularity based caching strategy for the future internet," in *ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT), 2016.* IEEE, 2016, pp. 1–8.

[106] Y. Li, H. Xie, Y. Wen, and Z.-L. Zhang, "Coordinating in-network caching in content-centric networks: Model and analysis," in *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on.* IEEE, 2013, pp. 62–72.

[107] J. Ren, W. Qi, C. Westphal, J. Wang, K. Lu, S. Liu, and S. Wang, "Magic: A distributed max-gain in-network caching strategy in information-centric networks," in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on.* IEEE, 2014, pp. 470–475.

[108] R. Hamza, K. Muhammad, Z. Lv, and F. Titouna, "Secure video summarization framework for personalized wireless capsule endoscopy," *Pervasive and Mobile Computing*, vol. 41, pp. 436–450, 2017.

[109] A. V. Vasilakos, Z. Li, G. Simon, and W. You, "Information centric network: Research challenges and opportunities," *Journal of Network and Computer Applications*, vol. 52, pp. 1–10, 2015.

[110] P. Kuppusamy, K. Thirunavukkarasu, and B. Kalaavathi, "A review of cooperative caching strategies in mobile ad hoc networks," *International Journal of Computer Applications*, vol. 29, no. 11, pp. 22–26, 2011.

[111] G. Lee, L. Han, Y. Park, J.-B. Lee, J. Kim, and H. P. In, "An energy-efficient routing protocol for ccn-based manets," *International Journal of Smart Home*, vol. 7, no. 1, pp. 143–152, 2013.

[112] E. M. Daly and M. Haahr, "Social network analysis for information flow in disconnected delay-tolerant manets," *IEEE Transactions on Mobile Computing*, vol. 8, no. 5, pp. 606–621, 2009.

[113] R. Fei, K. Yang, and X. Cheng, "A cooperative social and vehicular network and its dynamic bandwidth allocation algorithms," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on.* IEEE, 2011, pp. 63–67.

[114] D. Lee, J. Choi, J.-H. Kim, S. H. Noh, S. L. Min, Y. Cho, and C. S. Kim, "Lrfu: A spectrum of policies that subsumes the least recently used and least frequently used policies," *IEEE transactions on Computers*, vol. 50, no. 12, pp. 1352–1361, 2001.

[115] E. Chlebus and J. Brazier, "Nonstationary poisson modeling of web browsing session arrivals," *Information Processing Letters*, vol. 102, no. 5, pp. 187–190, 2007.