Master of Science in Computer Science and Engineering

Hybrid Caching with a Dynamic Forwarding Strategy in ICN

by

Ashim Dey

ID: 17MCSE003P

This thesis is submitted in partial fulfillment of the requirement for the degree of
MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering
Chittagong University of Engineering and Technology
Chattogram-4349, Bangladesh.

January, 2025

EXTRA PAGE DO NOT BINDCERTIFICATION

The thesis titled **Hybrid Caching with a Dynamic Forwarding Strategy in ICN** submitted by **Ashim Dey**, Roll No. **17MCSE003P** , Session **2017-2018** has been accepted as satisfactory in partial fulfillment of the requirement for the degree of **Master of Science in Computer Science and Engineering** on **23/01/2025**.

BOARD OF EXAMINERS

1. ____
Dr. Mohammad Shamsul Arefin — Chairman (Supervisor)
Professor, Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET)

2. ____
Dr. Asaduzzaman — Member (Co-Supervisor)
Professor, Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET)

3. ____
Dr. Abu Hasnat Mohammad Ashfak Habib — Member (Ex-officio)
Professor and Head, Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET)

4. ____
Dr. Muhammad Ibrahim Khan — Member (Internal)
Professor, Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET)

5. ____
Dr. Mohammed Moshiul Hoque — Member (Internal)
Professor, Department of Computer Science and Engineering
Chittagong University of Engineering & Technology (CUET)

6. ____
Dr. Md. Obaidur Rahman — Member (External)
Professor, Department of Computer Science and Engineering
Dhaka University of Engineering & Technology (DUET)

CANDIDATE'S DECLARATION

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Signature of the Candidate

____

Ashim Dey
ID: 17MCSE003P

Dedicated to,

my beloved parents,

Late. Shubhash Chandra Dey and Kanan Dey

my mentor and elder brother,

Sukanta Dey Amit

my lovely wife,

Annesha Das

my energetic son,

Awmayik Dreek Dey

for their love, endless support, encouragement, and sacrifices.

Acknowledgment

First and foremost, I would like to thank Almighty god for giving me the strength to complete this thesis. The satisfaction that accompanies the successful completion of this thesis would be incomplete without the mention of people whose ceaseless cooperation made it possible, and whose constant guidance and encouragement crown all efforts with success. I owe thanks to a number of people without whom this thesis would not have been possible. I want to thank my supervisor, Professor Dr. Mohammad Shamsul Arefin, for his continuous support and assistance. I am also grateful to my co-supervisor Prof. Dr. Asaduzzaman for his constant encouragement and stimulating ideas. I would also like to thank the members of my oral examination board for providing me with their valuable pieces of advice and feedback.

Finally, I am thankful to all the faculty members and staff of the department of CSE, CUET, for their support. My family deserves special thanks for their continuing love, support, encouragement, and sacrifices.

Abstract

In this era of huge data traffic, the information-centric network (ICN) is a switching strategy from conventional host-centric communication to a new data-centric one. In-network caching an integral part of ICN can reduce the shortcomings of today's location-based internet paradigm. As ICN has limited caching capability, the decision of optimal routing and content placement are the major challenges. In this study, we suggest a hybrid caching with a dynamic forwarding (*HCDF*) strategy which takes advantage of the off-path hash routing scheme and on-path caching scheme. To facilitate both caching schemes, a fraction of the caching space is allotted in each cache node for local caching. At the user-end segment, an estimation-based content placement (ECP) policy is designed using content popularity state and cache node position. This policy helps to cache popular content near the consumer's location using the shared cache space. At the server-end segment, the *HCDF* scheme dynamically selects the most appropriate routing strategy among three strategies (symmetric, asymmetric, and multicast) based on hop count and path stretch. To demonstrate the effectiveness of *HCDF*, extensive simulation on real-world internet topologies has been carried out. From our investigation, it is observed that the suggested approach exhibits an impressive performance in terms of cache hit, latency, and average link load compared to other standard hash routing and well-known on-path caching schemes.ICN; Hash routing; Hybrid caching; Content placement; Caching strategy.

Keywords: ICN, Hash routing, Hybrid caching, Content placement, Caching strategy.

সংক্ষিপ্তসার

বিশাল ডেটা ট্র্যাফিকের এই যুগে, তথ্য-কেন্দ্রিক নেটওয়ার্ক (ICN) হল একটি প্রচলিত হোস্ট-কেন্দ্রিক যোগাযোগ থেকে একটি নতুন ডেটা-কেন্দ্রিক যোগাযোগের কৌশল। এই ICN এর একটি অবিচ্ছেদ্য অংশ ইন-নেটওয়ার্ক ক্যাশিং যা ত্রুটিগুলি কমাতে পারে আজকের অবস্থান-ভিত্তিক ইন্টারনেটের দৃষ্টান্তে। যেহেতু ICN এর ক্যাশিং ক্ষমতা সীমিত, তাই সর্বোত্তম রাউটিং এবং বিষয়বস্তু বসানোর সিদ্ধান্ত প্রধান চ্যালেঞ্জ এই গবেষণায়, আমরা একটি ডায়নামিক ফরওয়ার্ডিং (HCDF) কৌশল সহ একটি হাইব্রিড ক্যাশিংয়ের পরামর্শ দিই যা অফ-পাথ হ্যাশ রাউটিং স্কিমের সুবিধা নেয় এবং অন-পাথ ক্যাশিং স্কিমেরও। উভয় ক্যাশিং স্কিমের সুবিধা পেতে, একটি ভগ্নাংশ স্থানীয় ক্যাশিংয়ের জন্য প্রতিটি ক্যাশে নোডে ক্যাশিং স্থান বরাদ্দ করা হয়। ব্যবহারকারীর শেষ সেগমেন্ট, একটি অনুমান-ভিত্তিক কন্টেন্ট প্লেসমেন্ট (ECP) নীতি ব্যবহার করে ডিজাইন করা হয়েছে বিষয়বস্তুর জনপ্রিয়তার অবস্থা এবং ক্যাশে নোড অবস্থা্নের উপর। এই নীতি শেয়ার করা ক্যাশে স্থান ব্যবহার করে ভোক্তার অবস্থানের কাছাকাছি জনপ্রিয় সামগ্রী ক্যাশ করতে সহায়তা করে। সার্ভার-এন্ড সেগমেন্ট এ, HCDF স্কিম গতিশীলভাবে সবচেয়ে উপযুক্ত নির্বাচন করে তিনটি কৌশলের মধ্যে রাউটিং কৌশল (সিমেট্রিক, অ্যাসিমেট্রিক এবং মাল্টিকাস্ট) হপ গণনা এবং পথ প্রসারিত উপর ভিত্তি করে. HCDF এর কার্যকারিতা প্রদর্শন করতে, বাস্তব-বিশ্বের ইন্টারনেট টপোলজিতে ব্যাপক সিমুলেশন করা হয়েছে। আমাদের তদন্তে দেখা গেছে যে প্রস্তাবিত পদ্ধতিটি ক্যাশে হিট, লেটেন্সি এবং গড় লিঙ্ক লোডের তুলনায় একটি চিত্তাকর্ষক কর্মক্ষমতা প্রদর্শন করে অন্যান্য স্ট্যান্ডার্ড হ্যাশ রাউটিং এবং সুপরিচিত অন-পাথ ক্যাশিং স্কিম থেকে।

কীওয়ার্ড: ICN, হ্যাশ রাউটিং, হাইব্রিড ক্যাশিং, কন্টেন্ট প্লেসমেন্ট, ক্যাশিং কৌশল

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, the Information-Centric Network (ICN) has acquired a lot of curiosity among researchers as an optimistic Future Internet (FI) design. The existing IP-based internet is built upon a host-centric paradigm. However, in ICN, transmission is dependent on the content itself rather than providers. Unlike conventional host-to-host communication, the specific location of information is not the main concern but the information itself is. In-network caching is the main attribute of ICN which enables users to receive their desired data timely and efficiently. This effective in-network caching feature is provided by equipping the routers with cache. These routers can then easily provide pre-cached data without interrupting the origin server. Caching is the preeminent practical approach that can be useful in any information-conformed technology and service. To reduce the data traffic and service latency in information-centric technologies and services, in-network caching can be an outstanding solution. Moreover, ICN can provide efficient and robust content retrieval by reducing origin server load and traffic redundancy. Attention towards the development of ICN has been growing day by day since its inception.

1.1 ICN Architecture

Many ICN-based architectures have been proposed towards future internet (FI) design as a result, for example, Data-Oriented Network Architecture (DONA)[3], Network of Information (NetInf)[4], Named Data Networking (NDN)[5], Content Centric Networking (CCN)[6], Pursuing a Publish Subscribe Internet Technology (PURSUIT)[7], etc [8]. Handling of request and response by an NDN router is shown in Fig. 1.1. Two categories of packet flow in ICN specifically 'Interest Packet' which represents details about the requested content and another 'Data Packet' which contains the requested data itself [9]. Obviously, the size of the

'Data Packet' is much larger than the 'Interest Packet'.

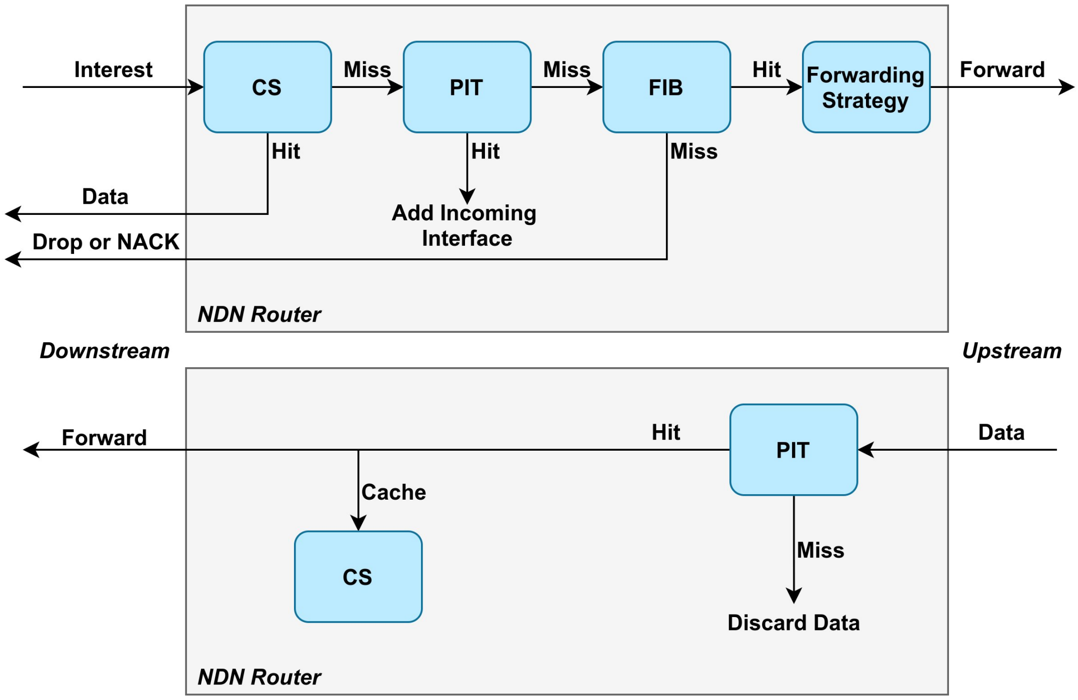


Figure 1.1: An NDN router's packet handling process[1][2]

There are common three classes of data structures available in ICN: Content Store (CS), Forwarding Interest Base (FIB), and Pending Interest Table (PIT)[10]. Forwarded contents are stored in CS which is a cache memory. The table named PIT maintains a trace of the still unsatisfied forwarded interest packets. Lastly, FIB maintains a table that stores forwarding information about the prefixes of known content. It enables the identification of the node holding the requested content. In fig.1.2, a CCN-based ICN architecture is depicted.

Two named packets are exchanged in receiver-driven ICN communication [11]. As shown in fig. 1.2, the interest packet, which consumers send to seek content, and the data packet, which carries the content and is returned by the data provider—which may be a cache node or the original producer. Three data structures are maintained at the data plane by each node: the Pending Interest Table (PIT), which retains forwarded interests that the data hasn't yet consumed, the Forwarding Information Base (FIB), which houses the outgoing interfaces for each named prefix, and the Content Store (CS), which stores incoming data in cache [12]. Initially, after being presented with an interest, the node searches the CS for corresponding data. The node searches for a similar request that is pending in the PIT if the content is not available locally. This means that the packet is rejected and the incoming interest interface is recorded in the PIT entry, which is then updated. The node searches the FIB for the outgoing interface(s) in the

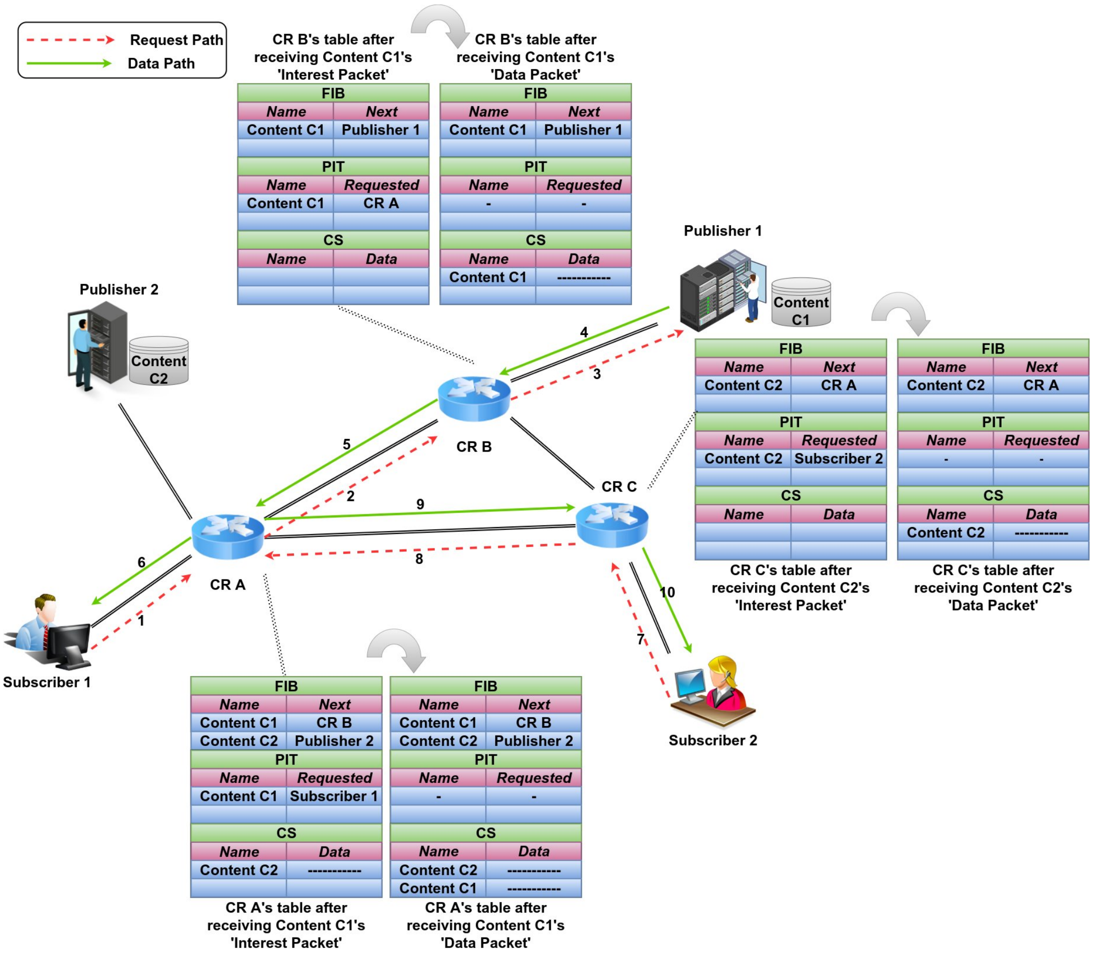


Figure 1.2: Detail representation of ICN architecture

event that the PIT matching is unsuccessful.

In the figure, 'Subscriber 1' generated a content request for 'Content C1'. At first, the content is searched in the intermediate Cache Router (CR): 'CR A' and 'CR B'. Upon cache miss, the interest packet is forwarded to the original content server 'Publisher 1' for 'Content C1'. During forwarding the data packet containing the requested content to the original requester 'Subscriber 1', one copy of the content is copied in both intermediate CRs. Similarly, Subscriber 2' generated a content request for 'Content C2'. At first, the content is searched in the intermediate 'CR C'. Upon cache miss, the interest packet is forwarded to 'CR A'. As content C1 is already cached in 'CR A' requested data is collected from here and forwarder towards the 'Subscriber 2'. During forwarding the data packet of the requested content to the original requester 'Subscriber 1', one copy of the content is copied in intermediate 'CR C'. In ICN, this caching ensures the fulfillment of future requests without hampering the original server itself.

In Information-Centric Networking (ICN), content is identified by its name rather than its location or the layer in which it resides. This content-centric approach allows us to handle caching at both the application layer and network layer without requiring strict separation between the two. However, it is important to address the differences in how content at these layers is managed for caching:

1. Application Layer Content:

    - Nature: Application-layer content typically refers to data such as media files, web pages, or other user-specific content that consumers directly request. The content is usually larger, and retrieval is often dependent on user behavior (e.g., a request for a video stream or a document).
    - Caching: In ICN, caching at the application layer focuses on storing content objects identified by their unique names. These objects are cached at various network nodes (routers or edge caches) based on the likelihood of repeated requests from consumers. The Estimation-based Content Placement (ECP) policy in our proposed scheme dynamically places content closer to consumers based on content popularity and request recency, allowing for efficient caching and delivery of application-level content.

2. Network Layer Content:

    - Nature: Network layer content typically refers to routing-related information or network control data that helps manage and forward traffic across the network. This includes information such as packet headers, routing tables, or forwarding information that ensures data reaches the correct destination.
    - Caching: Caching at the network layer deals more with metadata and network state information that helps in efficient forwarding and routing decisions. The Forwarding Interest Table (FIB), Pending Interest Table (PIT), and Content Store (CS) are key components in ICN architecture that store routing-related information. These data structures help ensure that content requests are handled efficiently and can be redirected to the appropriate cache or origin server.

To deal with both types of content, our approach ensures that both application layer content (e.g., media files or documents) and network layer content (e.g., routing metadata) are cached efficiently, but in separate storage spaces within the network nodes. While application layer content is cached based on user requests and content popularity, network layer content is cached in the form of forwarding information and metadata to ensure quick routing and forwarding of requests.

By differentiating the content types and using dynamic caching strategies that account for both content and network conditions, we ensure efficient data caching for all consumers. Additionally, the hybrid caching strategy in the HCDF approach allows the system to make adaptive decisions about where to store application-level content and network-level metadata, balancing the load on both cache storage and network resources.

1.2 CDN Vs. ICN

Content Delivery Networks (CDNs) and Information-Centric Networks (ICNs) are both crucial for efficiently delivering content over the Internet. While they share a common goal of improving content access speed and reducing network congestion, they differ fundamentally in their architecture, approach, and scalability. Understanding these differences is key to appreciating the advantages of ICN over traditional CDN architectures, especially in the context of emerging internet traffic patterns.

Content Delivery Networks (CDN): A CDN is an architecture designed to reduce latency and increase content availability by caching content at various geographically distributed servers. The main idea behind CDNs is to replicate content at several locations (edge servers) so users can retrieve the content from a server closer to them. This reduces the load on the origin server and improves access speed. CDNs rely on a host-centric approach, meaning that the network is primarily concerned with the location of content servers, and users access content by specifying the server location.

However, CDNs have certain limitations:

- Dependence on server locations: CDNs' efficiency depends on the location and availability of the servers. Users still need to know which server to contact, and the system's ability to optimize content delivery is limited by its reliance on fixed server locations.

- Redundancy and scalability issues: As traffic grows, CDNs need to replicate content across more servers, which can lead to redundant storage and inefficient resource use. The system is unsuited to dynamically adapt to varying network conditions or content popularity.

- Lack of flexibility: CDNs' architecture is not flexible enough to deal with content requests in a content-centric way. Content distribution is tied to specific server locations and doesn't inherently optimize content retrieval based on user demand or network conditions.

Information-Centric Networking (ICN): In contrast, ICN is a more advanced network architecture designed to address the limitations of traditional, host-centric approaches like CDNs. ICN emphasizes content rather than the location of servers, enabling more flexible, efficient, and scalable content delivery. In ICN, content is identified by its name (rather than the server's location), and the network itself is responsible for efficiently routing and caching content. This content-centric approach allows ICN to provide significant advantages over CDN in various ways:

- Content Naming and Retrieval: ICN allows users to request content by its name, rather than the location of the server that hosts it. This removes the dependency on fixed server locations and enable the network to fetch content from the most appropriate node in the network, whether that be an edge cache or an origin server.

- In-network Caching: A key feature of ICN is in-network caching, where routers cache content directly. This means that frequently requested content is readily available in the network, reducing latency and reducing the load on the origin server. Unlike CDNs, which replicate content across servers in specific locations, ICN can dynamically cache content across the network, making content more readily available without the need for redundant data storage.

- Dynamic Adaptability: ICN can dynamically adapt to changes in network conditions. By caching content closer to users and adapting the routing strategies (such as symmetric, asymmetric, and multicast), ICN significantly reduces latency and optimizes the use of network resources based on real-time traffic and content demand.

ICN outperforms CDN in several critical areas:

- Reduced Latency: By using in-network caching and dynamic forwarding strategies, ICN can reduce the time it takes to access content, as it can be retrieved from the closest cache rather than being served from distant origin servers.

- Improved Scalability: ICN's content-centric architecture is more scalable than CDNs, as it doesn't require continuous replication of content across a large number of servers. Instead, ICN can adjust dynamically to traffic demand, improving resource utilization and reducing network congestion.

- Efficient Content Delivery: ICN improves content delivery by utilizing a hybrid approach, combining both on-path and off-path caching strategies. This enables more efficient content placement in the network and ensures that content is available near the consumer's location without redundancy or unnecessary detours

- Utilization: ICN's ability to dynamically select the best forwarding strategy based on network conditions allows it to make better use of available network resources. This contrasts with CDNs, which are more rigid in their approach and can lead to network bottlenecks due to redundant content delivery.

In summary, while CDNs provide an efficient solution for content delivery by reducing the distance between the user and the server, ICN takes this a step further by removing the dependency on server location, leveraging in-network caching, and allowing dynamic adaptation to network conditions. These advantages make ICN a more robust, flexible, and scalable solution, especially in the context of future internet architectures and the growing demand for content delivery

1.3 Motivation

The shortcomings of conventional host-centric communication paradigms have been brought to light by the exponential expansion of data traffic in contemporary networks. These paradigms result in inefficient content delivery because they place more importance on the host's location than the content itself. By emphasizing the content rather than the source, Information-Centric Networking (ICN) shows promise as an alternative. In-network caching, the fundamental component of ICN, presents considerable difficulties with regard to content placement, routing tactics, and cache optimization.

The various requirements of dynamic and large-scale networks cannot be adequately met by the current caching technologies, such as on-path and off-path caching, which only provide partial answers. While off-path caching might result in higher latency and network overhead, on-path caching frequently produces redundant data storage. These drawbacks highlight the necessity of a hybrid caching strategy that minimizes the drawbacks of both methods while combining their advantages. This research attempts to close the gap between theoretical efficiency and real-world applicability in ICN architecture by putting forward a hybrid caching approach with dynamic forwarding.

Our major motivations to work in this area are,

- Traditional host-centric communication paradigms are inadequate to handle the exponential growth of data traffic, necessitating a shift to content-focused approaches like Information-Centric Networking (ICN).
- Existing caching strategies (on-path and off-path caching) are insufficient for dynamic, large-scale networks, as they either lead to redundant data storage or increased latency and overhead.
- This research bridges the gap by proposing a hybrid caching scheme with dynamic forwarding, combining the strengths of both caching strategies to improve efficiency and practicality in ICN design.

1.4 Importance

According to Cisco Global traffic forecast (2017-2023) [13], the number of devices connected to IP networks will be more than three times the global population. Moreover, in the era of social networking and OTT platforms, most of the bandwidth is consumed by enormous multimedia content. As reported by Cisco, total IP video traffic will be 82% of all IP traffic. To handle this huge traffic, researchers are trying to find a way to offload this huge traffic from the core network to cache local edge-caching nodes. One of the notable solutions is caching capable ICN. Figure 1.3 indicates huge data traffic in just one minute on internet.

According to estimates, total mobile network data movement will grow to an estimated 466 EB per month by the end of 2029 globally [14]. Another report indicates that by 2030, the monthly global market for mobile data traffic is expected to exceed 603.5 million gigabytes [15]. To handle these upcoming challenges ICN can be a game changer. It is not possible to identify the exact location of the content in the router node in the existing network system, but in ICN the accurate location of the content in the router can be easily identified. In ICN, it's the network's responsibility to fulfill the user's request from the best possible content location. Still, ICN is in the development stage, and many challenges have to be resolved.

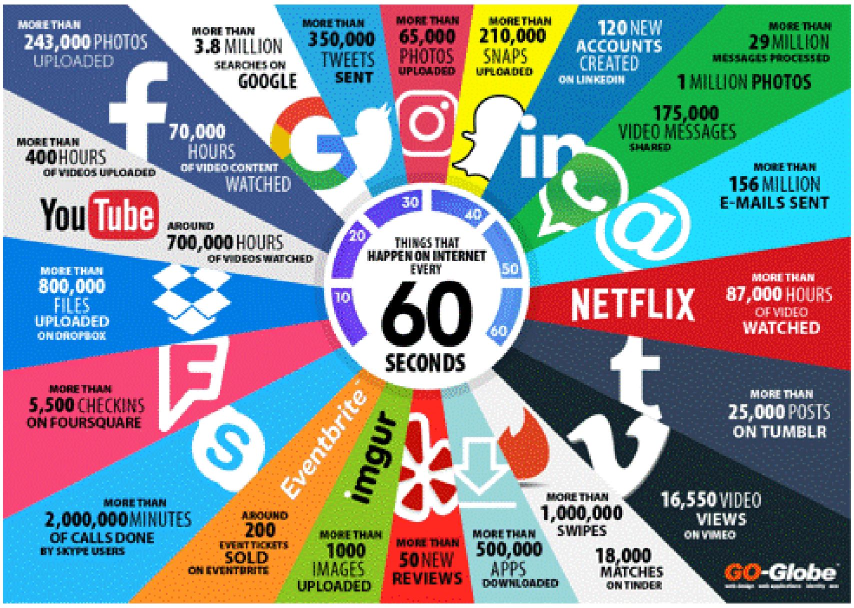


Figure 1.3: What happen in the internet in one minute?

There are various reasons why the suggested hybrid caching technique is important. By creating a system that adjusts to network dynamics, it first tackles significant shortcomings in ICN's existing caching techniques. This flexibility guarantees effective use of the limited cache space, speeds up the retrieval of material, and lowers latency—all of which are critical for contemporary network performance.

Additionally, the study advances the creation of reliable and scalable ICN systems that can be used in the real world. Through comprehensive simulations on real-world topologies, this work demonstrates higher performance, laying the groundwork for the implementation of hybrid caching techniques in a variety of network scenarios. The system is especially pertinent in meeting the demands of future internet architectures and the growth of data-intensive applications because of its capacity to balance cache hits, link load, and latency.

Some important real-world implications of this thesis are illustrated in the following,

- Enhanced Network Efficiency: The hybrid caching scheme reduces latency and link load while improving cache hit rates, directly benefiting real-world applications like content delivery networks, IoT, and streaming services.

- Scalability for Future Networks: By addressing dynamic network conditions, this approach supports scalable implementations in diverse topologies, making it suitable for large-scale deployments in future internet architectures.

- Practical Implementation Potential: The proposed scheme's adaptability and demonstrated performance on real-world topologies provide a solid foundation for integration into existing ICN frameworks, enhancing their robustness and practicality.

1.5 Challenges

Developing the hybrid caching scheme faced several challenges, including:

- **Optimal Cache Utilization:** Efficiently allocating limited cache space to maximize performance while minimizing redundancy.
- **Content Placement:** Designing a policy to determine where and when to cache content for maximum benefit.
- **Routing Strategy Selection:** Dynamically choosing between symmetric, asymmetric, and multicast routing strategies based on network conditions.
- **Balancing On-path and Off-path Caching:** Leveraging the strengths of both approaches while addressing their inherent trade-offs.
- **Scalability and Adaptability:** Ensuring the solution performs well across varying network topologies, sizes, and traffic patterns.

These challenges required innovative approaches, such as the Estimation-based Content Placement (ECP) policy and dynamic forwarding mechanisms, to ensure the hybrid scheme's effectiveness and efficiency.

1.6 Contributions

This research makes the following key contributions:

- **Hybrid Caching System:** Developed a caching mechanism that integrates on-path and off-path caching strategies by allocating fractions of cache space for local and authoritative caching.
- **Estimation-based Content Placement (ECP):** Introduced a novel policy that leverages content popularity, request recency, and hop reduction to dynamically place content closer to end users.

- **Dynamic Forwarding Strategy:** Designed a routing strategy that dynamically selects between symmetric, asymmetric, and multicast routing based on hop count and path stretch, optimizing cache hits and minimizing latency.

- **Comprehensive Evaluation:** Conducted extensive simulations on real-world topologies (GEANT, GARR, TISCALI) to demonstrate the scheme's superiority in terms of cache hit ratio, latency, and link load compared to standard on-path and off-path caching strategies.

By addressing these aspects, this research significantly advances the state of the art in ICN caching strategies, paving the way for more efficient and scalable future internet designs.

1.7 Organization of Thesis

The remaining of the thesis is organized as follows. Chapter 2 reviews existing caching strategies in Information-Centric Networking (ICN), including on-path caching, off-path caching, and hybrid caching techniques. Identifies gaps in current research that this thesis aims to address. Chapter 3 explains the proposed Hybrid Caching with Dynamic Forwarding (HCDF) strategy, including the hybrid caching-enabled system, the Estimation-based Content Placement (ECP) policy, and the dynamic forwarding mechanism. Describes the implementation details and operational architecture. Chapter 4 illustrates the simulation environment, experimental setup, and evaluation metrics. Discusses the performance of the HCDF strategy compared to existing caching methods. Chapter 5 summarizes the research findings, outlines the limitations of the study, and provides recommendations for future work.

Chapter 2

Literature Review

Network caching based on location can be classified into on-path caching and another is off-path caching[16][17]. The local features of the nodes on the content return path are used by on-path caching techniques to determine which nodes are suitable to store content copies. Any node outside the path where the server returns the content to the user can store content copies in off-path caching schemes. The literature review on ICN caching techniques is broken down into the following subsections for easier comprehension.

2.1 On-path caching

Regarding on-path caching, the corresponding content is cached along the same path the interest packet sent from the requester and which is received by the content origin [2]. In this, the request packet and content packet follow the same route. Leave Copy Everywhere (LCE) [18] which is the default and straightforward scheme of caching in ICN, caches the content in every node across the path. As every content is cached on each router along the route, this approach demands a great amount of space in the network for caching a single content. Both Leave Copy Down (LCD) as well as Move Copy Down (MCD) [19], try to reduce this redundancy to some extent. But LCD slows the process of pushing the freshly arrived content to the edge router. On the other side, MCD reduces the cache hit rate as it moves the replica of the content from the hit node to a single hop downstream toward the client. These strategies follow a deterministic caching mechanism. Some probabilistic caching mechanisms are also proposed where routers probabilistically make individual decisions on content caching. In ref.[20], authors proposed *ProbCache* which calculates cache probability depending on the cache ability of the path and approximated traffic cache serves per unit of time. This strategy caches content at a faster speed to edge nodes but with the price of a cache miss. With some modifications in ref.[21], they proposed an enhanced version of this algorithm called *ProbCache+*.

Many researchers take into account the centrality measures (degree, betweenness, closeness, reach) of a network topology while designing a caching strategy. These measures indicate the important caching nodes inside the network. WK Chai et al. [22] presented a betweenness centrality-based caching strategy named CL4M. Here, at the node owning the highest betweenness-centrality across the path, content is cached only. In [23], authors designed a caching scheme namely Centrality-measures based algorithm (CMBA) in which a suitable caching node is selected based upon the highest centrality-measures value (CMV). Here, CMV actually denotes the average value of the four centrality measures mentioned earlier. J. Pfender et al. [24] proposed Approximate Betweenness Centrality (ABC) which is a simple lightweight caching mechanism that utilizes approximate centrality information while caching data in the context of information-centric IoT. Normally, caching strategies based on centrality measures need substantial information on the topology. Authors claimed that ABC can show a similar reduction in content delivery latency requiring any global knowledge, setup, or communication overhead. In [25], authors proposed a caching scheme named *opt-Cache* which is an optimization-based in-network caching policy. In this policy, instead of focusing on a single factor during making the caching decision, it jointly considers the popularity of the content and the distance of the content from the origin. QN Nguyen et al. [9] proposed a beneficial caching scheme with a utility-based replacement policy in which not more than a certain number of copies for every content either as a full-object or partially is cached based on its request rate and data congestion.

Instead of using conventional homogeneous cache allocation, authors proposed a lightweight cache assignment technique based on topology information as well as content information in [26]. Further, they proposed an on-path caching mechanism hinged on the expected number of copies (ENC) and proved that this scheme guarantees content diversity with the capability of reducing content redundancy. In [27], authors proposed a novel chunk-based caching mechanism named Progressive Popularity-Aware Caching Scheme (PPCS). In the scheme, they considered entire-object caching for popular contents and partial-progressive caching for non-popular contents to avoid wastage of cache space in ICN. In [28], authors proposed PB-NCC to resolve the issues of recurrent cache replacement and irrational content allocation in conventional caching schemes in ICN. In this strategy, the caching scenario needs to be assembled with an NRS, which stores the locations of contents cached in ICN.

2.2 Off-path caching

In off-path caching, data packets can disseminate beyond the path which is not confined across the delivery path. That denotes the request packet and the content packet may follow a different route. One of the notable solutions proposed in the off-path caching domain is hash routing as it provides improved load balancing over network connections and caching nodes, better flexibility against traffic spikes as well as decreased inter-domain traffic. Hash-routing is an extensively used approach regarding web caches. Five different hash-routing strategies were presented by Saino et al. [29]. Here, authors performed in-network caching without depending upon network routers to manage information of per-content state. In [30], authors split large domains into clusters and applied hash-routing in the subgroup of the nodes of every cluster rather than applying it in the entire domain as the original hash-routing strategy. In another work, K. Hasan et al.[31] came up with a cluster-based caching solution in ICN to reduce the on-path and hash caching shortcomings. In both these approaches, authors tried to restrict traveling the requests far in the network in order to reduce network overhead.

In another work, H. Jung et al.[32] designed an innovative off-path caching technique for prompt data recovery. This proposed off-path technique is named Geographic information-based Hash Routing (GHR) for ICN. The fundamental approach of GHR is to hash data names straightaway into geographic values to determine the nearest data store. In [33], proposes a hash-based routing architecture on ICN in which content routers and namespaces are grouped using the hash function. In this proposed architecture, hash-based clusters are created in every domain applying globally shared hash functions including the global namespace is distributed to various clusters. In [34], authors proposed online caching along with a cooperative forwarding scheme. To obtain an online coordinated caching solution, authors formulated the caching complication as a Ski-Rental problem along with consistent hashing. Authors in [35], presented a caching scheme namely spatially dispersed caching (SDC). Here, a distinctive binary ID is allotted in every router and then the cache target of content is limited at every router whose hash value co-occurs with the router ID.

2.3 Hybrid caching

Instead of utilizing a single caching technique in a single network, a few hybrid techniques combining multiple caching strategies have also been exploited in the field of ICN. H. Wu et al. [36] proposed *Hcache*, a hybrid caching strategy in which for different regions, they utilize different caching techniques. Here, they applied deterministic caching in edge areas to minimize the delay of access from the user's point of view, and probabilistic caching in the core area to lessen inter-domain traffic from the ISP's side and increase the cache diversity. Some other hybrid caching techniques have been also experimented with trade-offs between on-path caching and off-path caching in the field of ICN. In these techniques, researchers have tried balancing the performance of cache hits and scalability by mixing the features of on-path and off-path caching techniques. In [37], authors proposed a hybrid caching strategy named EDOP which couples off-path caching with edge caching. Here, the authors tried to improve the average retrieval delay.

2.4 Research Gap

Though many caching strategies have been proposed, ICN is still facing hardship in selecting a suitable router to cache during content transmission. Selection of a suitable router can minimize the distance between the clients and servers as well as deliver with minimal latency in data transmission with little congestion [38]. On-path caching approaches can enhance network performance to a certain extent. Though the extensibility of all the on-path caching is substantial, it decreases cache hits because of unnecessary caching of contents. On the flip side, off-path caching strategies store content that is not in the path of the request to the server. That means off-path caching involves storing content at locations that are not directly on the path of the request, it may lead to increased latency when retrieving cached content. The request may need to traverse additional hops to access the cached data. This situation increases the latency of content retrieval due to the extra hops required. So, we proposed a strategy namely HCDF that balances the two approaches and takes the best advantages out of these.

The key research questions we are investigating in this work are-,

- "**RQ1:** Which data packet to evict when cache space runs out?"
- "**RQ2:** Which node to select for caching to get the maximum benefit?"
- "**RQ3:** Which routing strategy to follow to get the maximum cache hit and minimum link load and latency?"

In the subsequent subsections, we comprehensively discuss the way of addressing these research questions.

Chapter 3

Methodology

In this section, at first, the basic hash routing functional architecture and the proposed system model are presented. After that, our content placement policy is described. Finally, we present our hybrid hash caching with dynamic forwarding strategy.

3.1 Basic Hash Routing Operational Architecture

Hash-routing mechanism can be easily incorporated with existing ICN, performing some modification as proposed in [29]. Normally, hash-routing schemes fall in the domain of off-path caching. In the operational architecture, both the edge routers and the core routers are required to perform a hash function that decides both the placing of newly arrived content and the routing process of the content request. Once an edge router is requested for content, it computes the hash of the content ID and re-routes the request to the corresponding cache node. If a cache hit occurs, the requested content is advanced towards the client, or else the request is redirected to the native server. The benefits of hash-routing approach can be summarized in three main points. Firstly, this approach restrains replication of the same content in the cache nodes. Secondly, without any lookup, the content request can be forwarded by edge routers straight to the responsible cache node. Thirdly, a hash function is naturally capable of scattering popular contents without any further communication overhead which results in great scalability.

In fig. 3.1, the operational architecture of hash-routing is presented. When a request from 'User 1' for any content arrives at edge router 'ER1', the edge router computes the hash and finds the responsible node. Then, 'ER1' forwards the request to core router ‘CR3’ which is responsible for caching the content via ‘CR1’ and ‘CR2’. If the content is present in ‘CR3’, content is put back to the requested user following the same path, otherwise the request is forwarded to the origin server ‘Server 1’ via ‘CR5’ and ‘ER4’. Then, content is forwarded from the server ‘Server 1’ to the requested user ‘User 1’ following the same path through it was requested in case of cache miss. While returning content through ‘CR3’, a duplicate of the content is also created and saved in the cache. This hash routing is known as ‘Symmetric’ routing as the request path and the data path follow the same route.

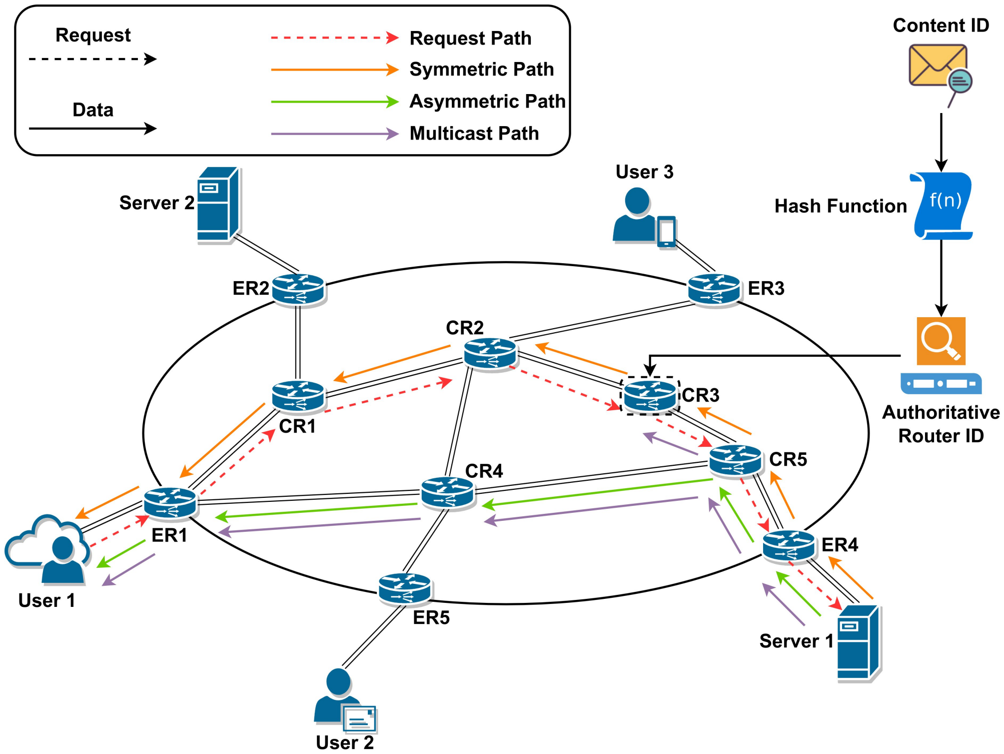


Figure 3.1: Hash-routing operational architecture.

However, in the ‘Asymmetric’ routing scheme, while forwarding the content from the source server to the requested user, routing follows the shortest path which in this case is via CR4. Here, the content can not be replicated in the responsible caching node. So, in this scheme, the request path and the data path may follow different routes. To avoid the replication problem, the ‘Multicast’ routing scheme comes to light. In this scheme, while forwarding the content from the server node to the request node following the shortest path, one replica of the content is also forwarded to the responsible cache node. It can be observed that during multicast while forwarding the requested data packet from ‘Server 1’ to ‘User 1’, one copy is also forwarded to cache node ‘CR3’ via ‘CR5’.

It can be noted that if the node responsible for caching the content lies on the shortest trail between the requester and the content provider, then all three schemes (asymmetric, symmetric, multicast) behave uniformly. Among the three basic schemes, symmetric and multicast hash-routing scheme produces high cache hits as these schemes ensure that content will be replicated to the authoritative cache node. But these schemes have to pay the cost, in terms of extra hops resulting in extra link load in the network. As a result, depending on the network topology, the performance of both symmetric and multicast degrades in terms of link load and delay. In contrast, asymmetric hash-routing stages lower effectiveness regarding cache hit. This strategy never ensures the caching of requested content in the responsible node as content is routed following the shortest path and the responsible node may not fall on that path. As a result, the performance of this scheme in terms of link load and delay greatly varies with topologies. One of the shortcomings of these off-path caching schemes is that they demand several extra hops while detouring the request as well as content packets.

3.2 Hybrid Caching with a Dynamic Forwarding (HCDF) strategy

Intending to reduce the negative impact of detouring, the proposed HCDF strategy is presented here. Three key features of the HCDF scheme are:

1. **Hybrid caching-enable system**
2. **Estimation-based Content Placement (ECP)**
3. **Dynamic Forwarding**

In the following, the three key features are briefly described.

3.2.1 Hybrid caching-enable system

Here, we have incorporated off-path caching with on-path caching to take advantage of both caching strategies by allotting a fraction of the caching space in each cache node for local caching. In our system, we propose some modifications to the basic hash routing model. Instead of allotting the whole cache space for mapping with the hash function, a fraction of the caching space in each cache node is reserved for local caching. This fraction of space can be predefined by the network administrator. The best space distribution depends on multiple factors including network size, available cache space, number of content, content popularity, content request pattern, network congestion, etc. It is practical to keep local cache space minimum. High local cache allotment will reduce the cache hit, as the authoritative cache will not be able to satisfy a large amount of requests. In the proposed scheme, each cache router has two fractions of storage, one is an authoritative cache and another is a local cache. The local cache is allotted for on-path caching and the authoritative cache is allotted for hash caching. Furthermore, considering the authoritative cache node as the median of the request path, we have divided the path into two segments namely the user-end segment and the server-end segment which is presented in Fig. 3.2.

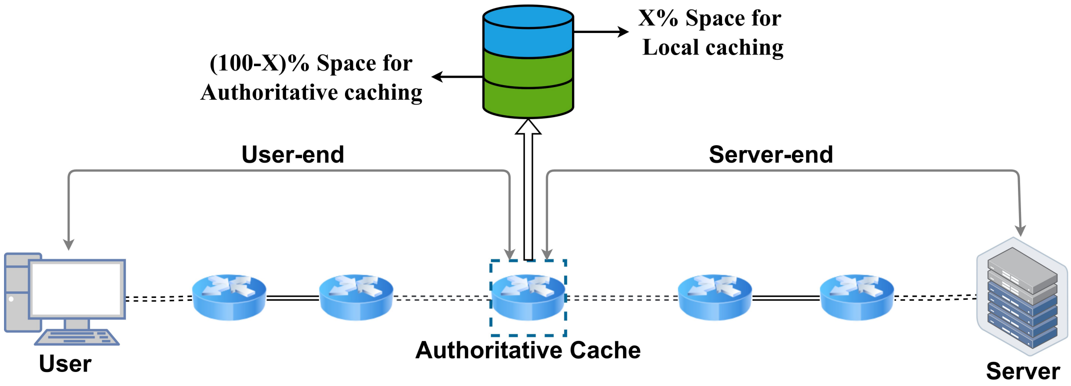


Figure 3.2: Overview of the proposed method.

Moreover, each cache node will maintain a Requested Packet Information (RPI) table. RPI table is added to all nodes to keep track of some basic information about the interest packet. RPI Table has three fields: content ID, number of requests, and last request time stamp. While an interest packet appears at a node, RPI table will update the number of requests and the last requested time of the respective content on that node. The detailed structure of an RPI table is presented in the Fig. 3.3.

Additionally, we reform the format of the data packet and interest packet to accommodate our proposed content placement policy Estimation-based Content Placement (ECP) in the network which is presented in Fig. 3.4. Two extra fields: *maximum value of estimation factor* as well as the *candidate node ID* inside the interest packet and one extra field *cache node ID* in the data packet are added.

3.2.2 Estimation-based Content Placement (ECP)

ECP is designed in such a way that it can move the popular content at the user-end segment towards the end user based on content request frequency, request recency, and cache node position. In the system, we propose to allot a fraction of the caching space in each cache node reserved for local caching. One of the key motivations is to advance popular content to the edge of consumers and

RPI Table		
Interest Content ID	**No. of Requests**	**Latest Request Timestamp**
1	**...**	**...**
. .	**...**	**...**
k	**...**	**...**

Figure 3.3: Structure of RPI table.

Interest Packet
Content ID
Maximum Value of Estimation Factor
Candidate Node ID
Other

Data Packet
Content ID
Data
Cache Node ID
Other

Figure 3.4: Structure of the packets.

build a distributed cache placement strategy using this space. So, an efficient and expandable cache placement technique namely Estimation-based Content Placement (ECP) is designed. The requested packet state and cache node position both are key factors in deciding content placement. The proposed ECP scheme takes into account content popularity, path reduction, and request recency. In this scheme content placement decisions are made based on the following three factors:

1. **Request Frequency (f)**: Request frequency indicates content popularity. That means how many times the content request has been received by a node. f is calculated using the following Eq. 3.1,

$$f = \frac{a_i^k}{a_i} \tag{3.1}$$

where a_i^k denotes the amount of requests arrived demanding content k at node i. a_i denotes the total amount of requests arrived at node i. This information can be obtained from RPI table easily.

2. **Request Recency** (r): Request frequency is not enough to estimate content popularity because popularity can change over time. Request recency r indicates the freshness of the content request. It can be calculated from the following Eq. 3.2,

$$r = \frac{1}{t^k_{in} - t^k_{io}} \tag{3.2}$$

where t^k_{in} denotes the current or new timestamp for the request demanding content(k) at node i and t^k_{io} indicates the previous or old timestamp for the request of content(k) at node i which could also be obtained from RPI table.

3. **Hop Reduction** (h): Hop reduction is an important factor during deciding the cache node. Popular content should be stored on the network edge. The hop reduction value depends on the position of the content requester, current cache node, and content provider. Equation 3.3 denotes hop reduction.

$$h = \frac{d^p_c - d^i_c}{d^p_c} \tag{3.3}$$

where d^p_c denotes the gap between the content consumer c and the content provider p. d^i_c implies the gap between the content consumer c and the current visiting node i.

Based on these three variables, the estimation factor e_k is calculated using the following Eq. 3.4. Before calculating the estimation factor, variables are normalized.

$$e_k = f \times \alpha_1 + r \times \alpha_2 + h \times \alpha_3 \tag{3.4}$$

where α_1, α_2, and α_3 are adjustment factors and $\alpha_1 + \alpha_2 + \alpha_3 = 1$. These adjustment factor values can be tweaked according to network demand. In ECP, when the data packet of content k traverses back from the content provider, one replica of the content k will be reserved in the cache of a node having a maximum e_k value for content k. Details of the proposed *ECP* scheme are shown in Algorithm 1.

3.2.3 Dynamic Forwarding

The proposed HCDF scheme dynamically selects the most appropriate routing strategy among three strategies (symmetric, asymmetric, and multicast) based on

Algorithm 1 Estimation-based Content Placement (ECP)

Input : Interest packet for content k, RPI Table , Data packet for content k

Initialize : Maximum Estimation factor, $e_k = -\infty$; Candidate node ID, $candidate_{id} = null$; Cache node ID, $cache_{id} = null$

for each node i one hop towards provider of content k **do**
    update RPI table of node i
    search content at node i
    if no cache hit at node i **then**
        calculate estimation factor at node i, e_k^i
        if $e_k^i > e_k$ **then**
            $e_k \leftarrow e_k^i$ and $candidate_{id} \leftarrow i$
        end if
    else
        fetch data packet out of the cache of the hit node
        Cache node ID , $cache_{id} \leftarrow candidate_{id}$
        for each node j one hop towards requester of content k **do**
            if $cache_{id} == j$ **then**
                store copy of content k at cache of node j
            end if
        end for
    end if
end for

hop count and path stretch. In this subsection, our proposed dynamic forwarding strategy is presented keeping in mind the two segments that are divided based on the placement of the authoritative cache node: User-end Segment and Server-end Segment.

User-end Segment: Segment from user to authoritative cache is considered as the user-end segment. In this segment, when a request for content arises by a user, the request is forwarded to the authoritative cache. While crossing each intermediate router towards the authoritative cache, this strategy looks up in the local cache of each router for the requested content. Besides, it updates the RPI table of each crossing node according to content ID. If a cache hit happens in any of the intermediary routers or the authoritative cache itself, ECP caching scheme will be performed. That means while forwarding the content from the cache hit node, it stores a replica of that content in the local cache of the node which has the maximum estimation factor towards the requested node. So, the more the content becomes popular, the more replicas of that content in the local caches will increase. Thus, it ensures moving the popular content from the core network towards the edge of the network. Figure 3.5, presents the complete process of the user-end segment.

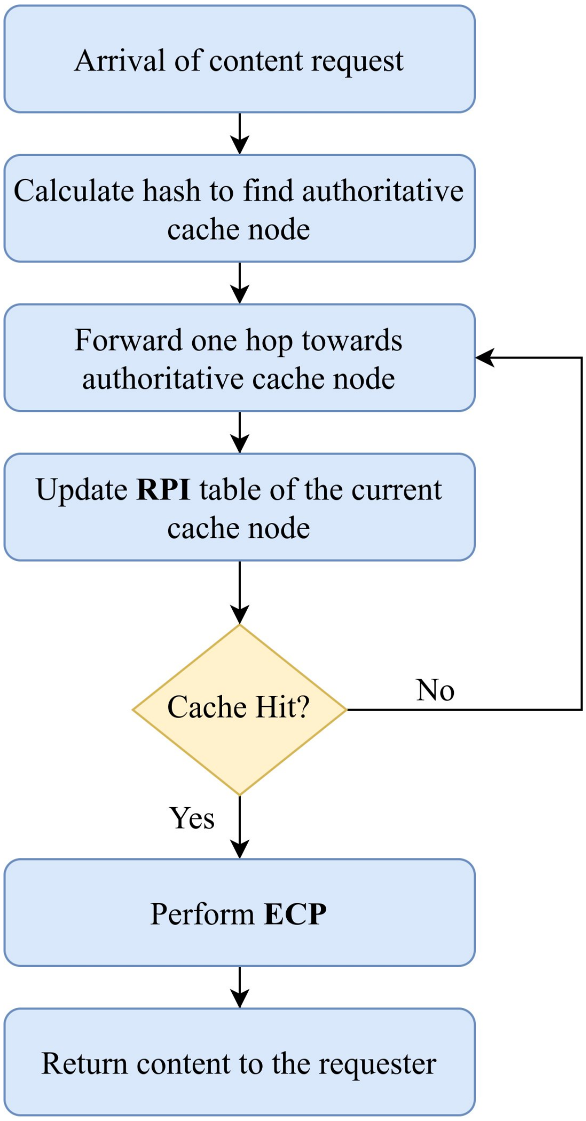


Figure 3.5: Flowchart of the User-end operation.

Server-end Segment: This segment represents the rest of the path left after the cache node till the origin server. Content request is seen in this segment if any cache hit does not occur in the previous user-end segment. Figure 3.6, presents the step-by-step process in the server-end segment. Surely, in this segment cache hit will occur either in an intermediate node's local cache or in the origin server. After the cache hits, we have to decide which routing algorithm to follow among symmetric, asymmetric, or multicast to return the content to the requested user. At first, the total hops required to perform symmetric and multicast routing are calculated separately. If the total hops required for symmetric is not more than multicast, symmetric routing will be performed. That means the content is forwarded to the user following the same request route via the authoritative cache node and while returning, a copy of that content is stored in the authoritative

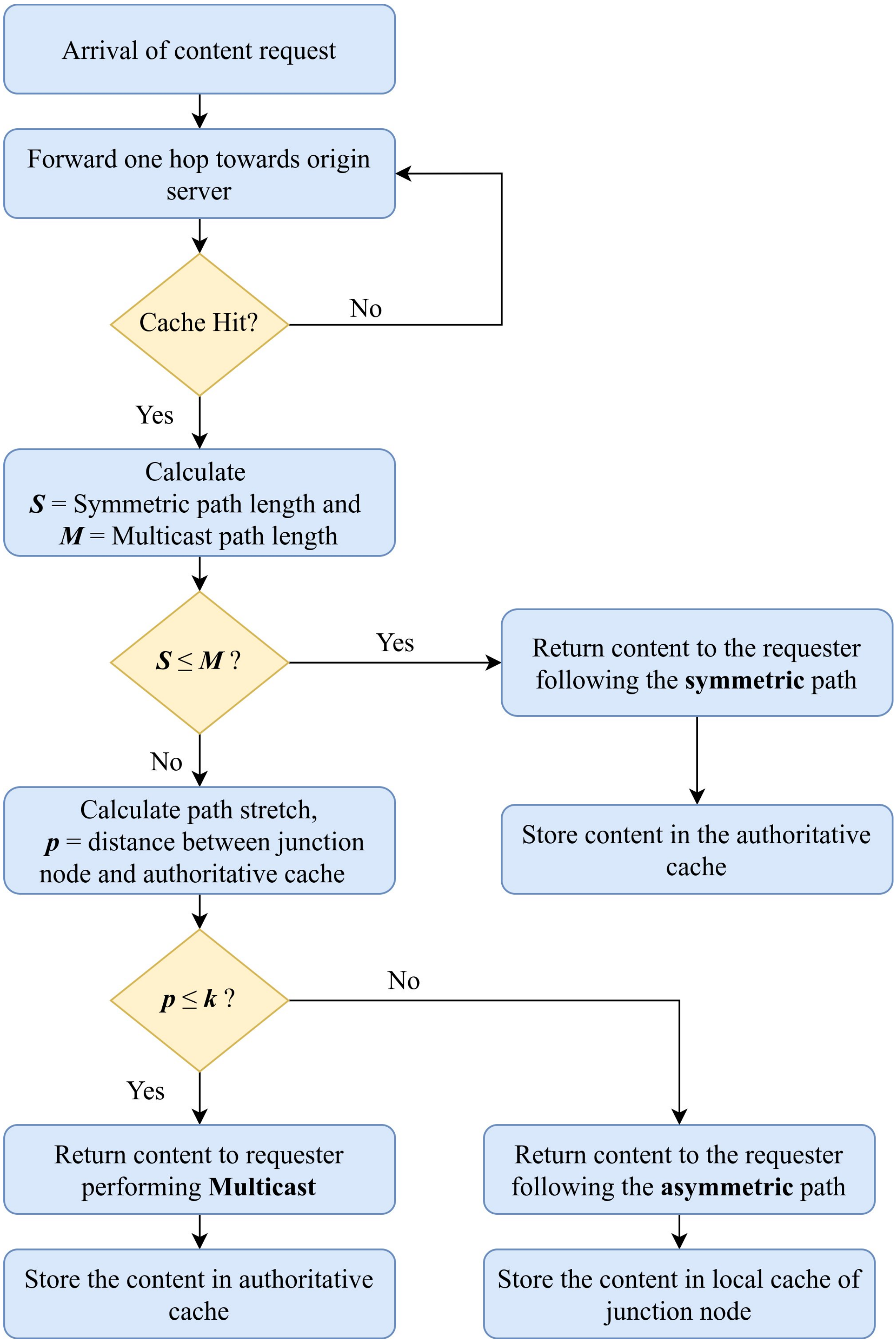


Figure 3.6: Flowchart of the Server-end operation.

cache. Otherwise, content is redirected to the user following the shortest path (asymmetric) from the hit node to the user. In this case, where the replication of the content will be performed, it depends upon the path stretch from the junction node to the authoritative cache node. The junction node is the node from which two paths including one path towards the user and the other path towards the authoritative cache node split. For instance, in Fig. 3.1, 'CR5' is the junction node for the content request from 'User 1'. If the path stretch from the junction node to the authoritative cache node is less than a predefined fraction $\boldsymbol{k}$ of the network's diameter $\boldsymbol{D}$, multicast is performed. That means one copy of that content is forwarded to the authoritative cache. Otherwise, content is simply stored in the local cache of the junction node. This step ensures two things. First, prohibits performing expensive multicast operations if it results in a high link load over the network, and secondly, in case of no multicast operation, caches the content in a node with a high degree which is also nearer to the authoritative cache. So, if again a request is generated by the user for that content, a cache hit occurs in the junction node rather than the remote server.

Chapter 4

Results and Discussions

We have performed extensive simulations to find the efficiency and feasibility of our proposed HCDF scheme. Simulations are performed in different network conditions. The details of our simulation are described in the following subsections:

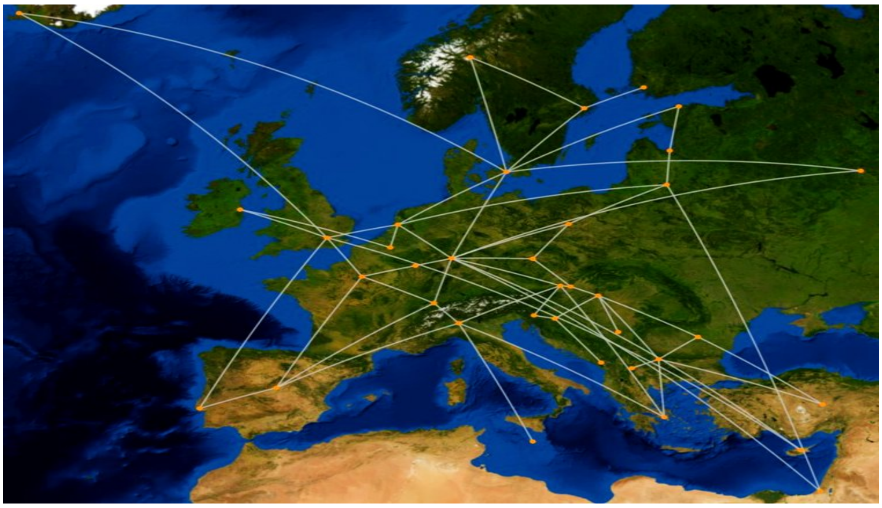
4.1 Simulation Environment

As a simulator, we have chosen *Icarus* (version 0.7.0) which is a Python-dependent caching simulator in ICN [39]. *Icarus* is publicly available on GitHub. *Icarus* is not compelled to any particular ICN framework, rather it allows its user to assess caching schemes in any ICN configuration along with a wide range of functionalities. Customized implementation of new strategies with fast and large-scale experiments is the main feature of this simulator. The CPU is Intel Core i5-4210U, 1.70GHz $\times$ 4, 8GB of RAM, Intel HD Graphics 4400 (HSW GT2), and the operating system is Ubuntu 18.04.6 LTS.

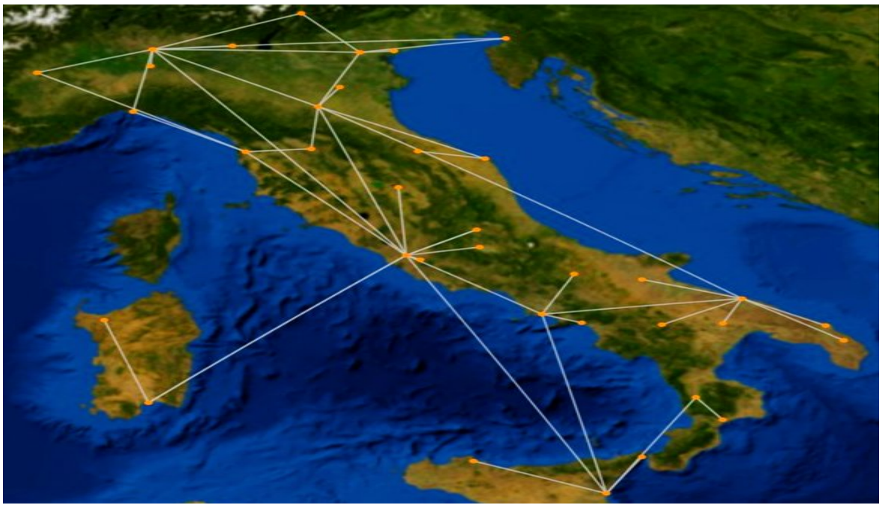
4.2 Experimental Setup

To compare our scheme, we have considered well-known on-path caching schemes as well as standard off-path caching schemes. For performance analysis simulations have been done on real-world internet topologies including GEANT (Europe), GARR (Italy), and TISCALI (Europe) as in Fig. 4.1.

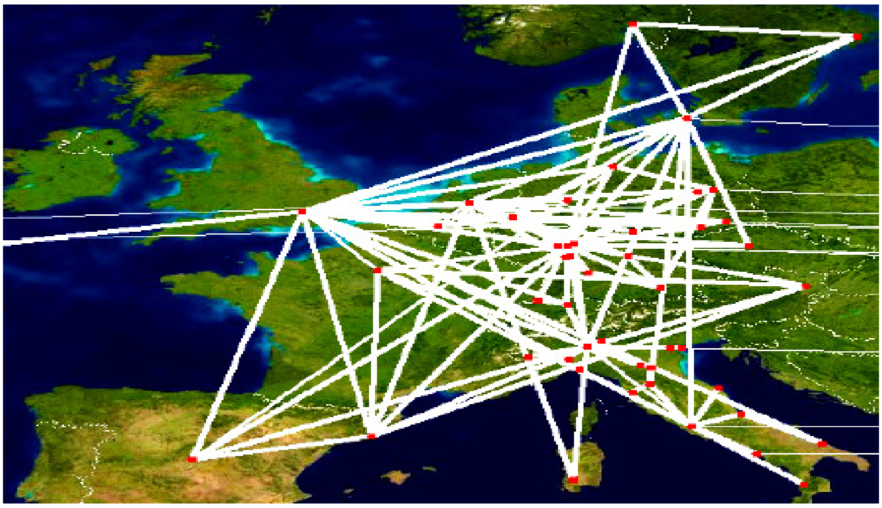
GEANT and GARR are collected from the work of the internet zoo topology [40] and TISCALI is provided by Rocketfuel project [41]. GEANT is a European research network connecting several countries as well as regions throughout the globe by high-speed links. This topology is considered as 13 producers, 8 consumers, and 32 routers with caching capability. GARR is a national computer



(a) GEANT with 53 nodes [40]



(b) GARR with 61 nodes[40]



(c) TISCALI with 240 nodes[41]

Figure 4.1: Selected topologies used in performance evaluation

network used by universities and the Scientific Research Institute of Italy consisting of 13 producers, 21 consumers, and 27 routers with caching capability. TISCALI is a large network of pan-European commercial ISPs with 44 producers, 36 consumers, and 160 routers performing as in-network cache. TISCALI is comparatively larger than the other two network configurations.

The proposed strategy's performance has been analyzed and contrasted with other existing caching strategies regarding average link load, cache hit, and latency varying different parameters like content popularity, cache size, etc. The proposed scheme distributes the total caching space into local caching and authoritative caching for each caching-capable node. In our experiments, 10% space was allotted for local caching and the rest space was allotted for authoritative caching. The simulator parameter used in our experiment is presented in Table 4.1.

Table 4.1: Parameters of Simulation.

Parameter	Value/Description
Simulator	Icarus (Python Based)
Topologies	GEANT,GARR,TISCALI
Cache Policy	LRU
Number of Content Objects	2×10^5
Number of Warm-up Requests	2×10^5
Number of Measured Requests	4×10^5
Request Rate	10 requests/second
Popularity Model	Zifp's Law, $\alpha \in$[0.6, 0.8, 1.0, 1.2]
Network Cache Ratio	C= [.0025, 0.005, 0.025, 0.05, 0.1]
Request Size	150 bytes
Content Size	1500 bytes
Intra Domain Link delay	2 millisecond (ms)
Inter Domain Link delay	32 millisecond (ms)

For cache replacement, the LRU strategy is utilized on all strategies. LRU is considered to be the most effective content replacement policy. The same caching capability was assigned to all routers. A total of 2×10^5 contents is used to populate the network and each content is considered to be 1500 bytes. The arrival of the consumer's request follows the Poisson process at a rate of 10 requests/second and content popularity is modeled on Zipf's distribution (α). Each request packet is considered to be 150 bytes which is 10 times smaller than the data packet. The parameter skewness of Zipf distribution, α ranges from [0.6-1.2] in our experiments. 2×10^5 request was generated to warm up the network and following 4×10^5 requests were used for evaluation. In the table network cache ratio (C) reflects the aggregated amount of network caches as a portion of the whole content occupants and its considered range is [.0025, 0.005, 0.025, 0.05, 0.1]. Each time, an average of three experiments is considered for evaluation.

4.3 Evaluation Measures

Here, we examine how choosing various factors influences the implemented content caching techniques in ICN via a simulated experiment. Three standard evaluation matrices including cache hit ratio, latency, and link load are considered. Details of these metrics are discussed in the following.

- **Cache Hit Ratio**: If a desired content is discovered in a cache node, it is considered as *cache hit*. Otherwise, it is a *cache miss*, and then the content is retrieved from the origin server. A cache hit is referred to as the segment of requested content served by the cache and is calculated from the following Eq. 4.1:

$$Cache\ Hit\ Ratio = \frac{N_{cache}}{N_{cache} + N_{server}} \tag{4.1}$$

  In the equation, N_{cache}, indicates the gross number of times the request is served by the network cache and N_{server} denotes the number of times the request is satisfied by the origin server.

- **Latency**: Latency defines the average time that a content requester has to wait from requesting content to actually getting the content. Average latency could be defined by the below Eq. 4.2:

$$Latency = \frac{\sum_{i=1}^{N} t_i}{N} \tag{4.2}$$

  where N indicates the overall number of requests generated in the network and t_i is the time total time taken from sending to receiving a request i.

- **Link Load**: We have evaluated the internal or intra-domain link load of the network using the following Eq. 4.3:

$$Link\ Load = \frac{\sum_{j=1}^{L} l_j}{L} \tag{4.3}$$

where, l_j indicates the link load of link j in per unit time and L means total number of link in the network.

4.4 Performance Evaluation

In this portion, we investigated the findings of the simulation in the same network environment and gave an analysis. We have scrutinized the performance of both on-path caching schemes (CL4M, ProbCache, LCE, LCD) and off-path such as well-known hash routing (HR) caching schemes (Symmetric, Asymmetric, Multicast) with our proposed method. We have also considered different performance indexes, such as latency, cache hit ratio, and link load to examine the performance of our proposed strategy in contrast with other well-known strategies.

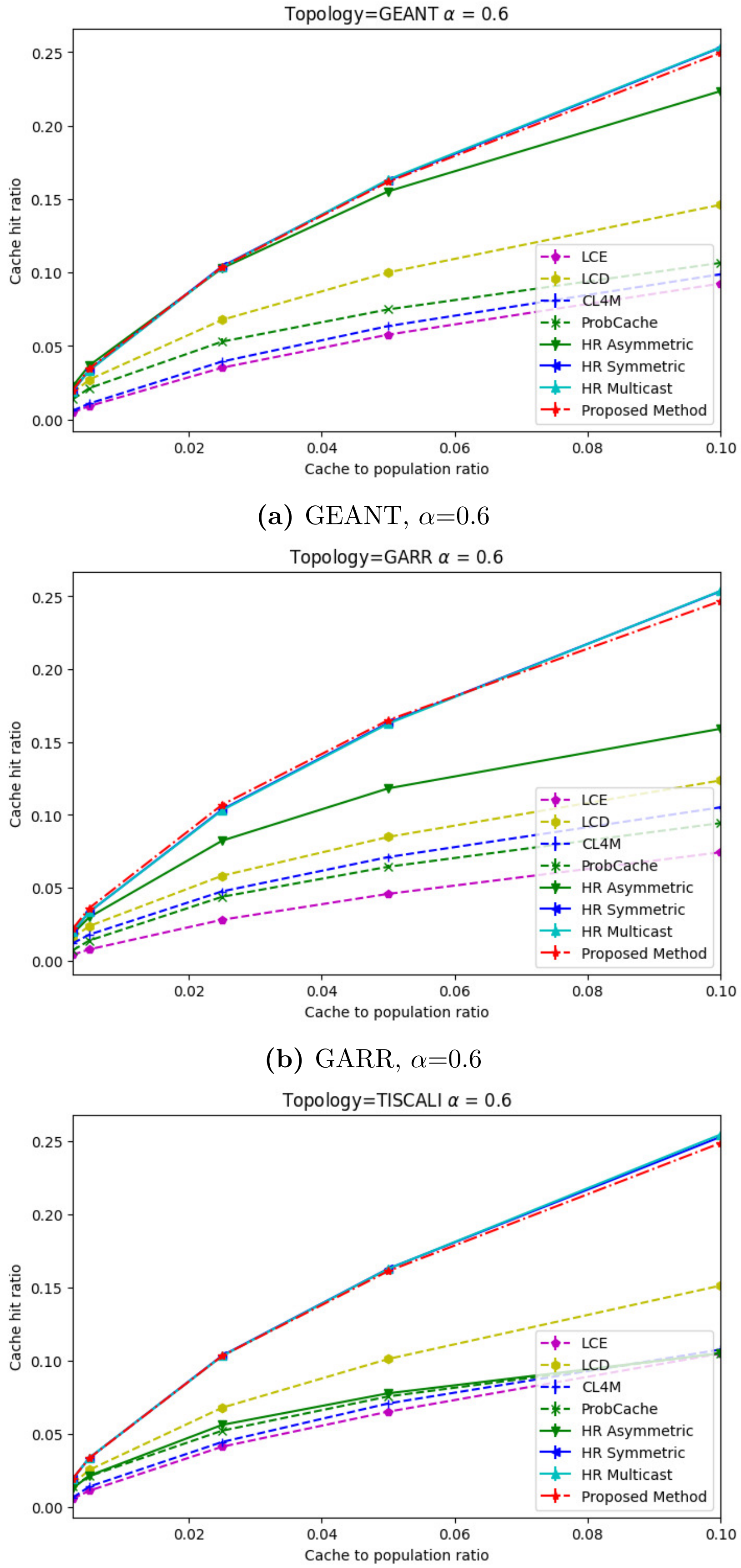
4.4.1 Impact on Cache hit ratio

In evaluating the performance of ICN, the cache hit is considered to be the most significant metric. If additional requests are being satisfied by intermediate cache routers, then the cache hit ratio will be higher reducing the load on the source. Soaring cache hit ratio also reflects, the low repetitiveness of cached content in the network [17].

Figure 4.2 (a, b, c) demonstrates the outcomes attained in three topologies: (GARR, GEANT, and TISCALI) respectively in terms of different cache to population ratios (C). In this segment of the investigation, skewness (α) is assigned to 0.6. Here, Among off-path caching schemes HR Symmetric and HR Multicast, outperform all on-path caching schemes which are expected. The proposed scheme exhibits higher and quite similar performance on all three topologies. One of the reasons is that these schemes ensure content will be cached in an intermediate node, which increases the possibility of a cache hit in the following content request. Only HR Asymmetric performs lowest among other off-path schemes especially in large network configurations (TISCAL) as this scheme does not ensure to always cache the content other than the source.

A similar cache hit is observed keeping constant the cache to population ratio (C) at 0.0025 in Figure 4.3 (a, b, c) on three topologies: (GARR, GEANT, and TISCALI) respectively in terms of different skewness (α).

The cache hit rate of the three topologies with fixed skewness, α=0.6, and fixed cache size ratio, C=0.0025 is presented in Fig. 4.4. This figure reflects that our proposed scheme performs significantly well on both small and large network configurations, especially in GARR topology. In GEANT, the proposed approach performed slightly low compared to HR Asymmetric.

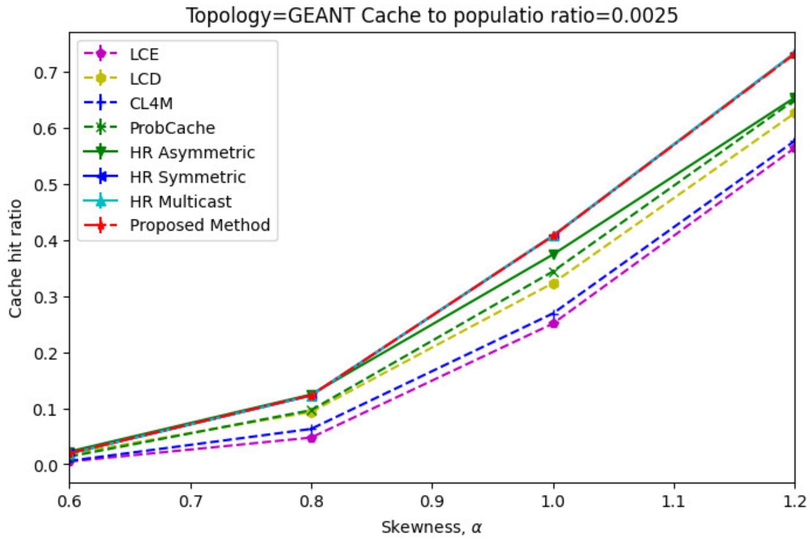


(a) GEANT, α=0.6

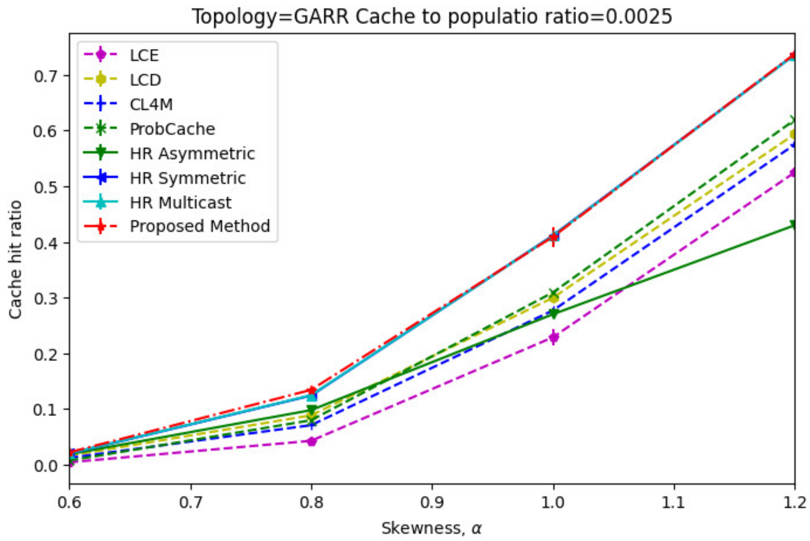
(b) GARR, α=0.6

(c) TISCALI, α=0.6

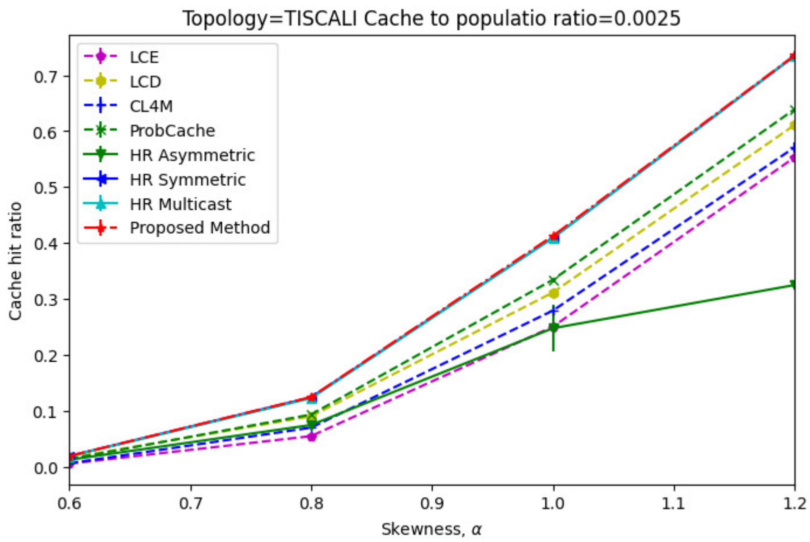
Figure 4.2: Performance of cache hit with different cache size ratio (C)



(a) GEANT, C=0.0025



(b) GARR, C=0.0025



(c) TISCALI, C=0.0025

Figure 4.3: Performance of cache hit with different popularity skewness (α)

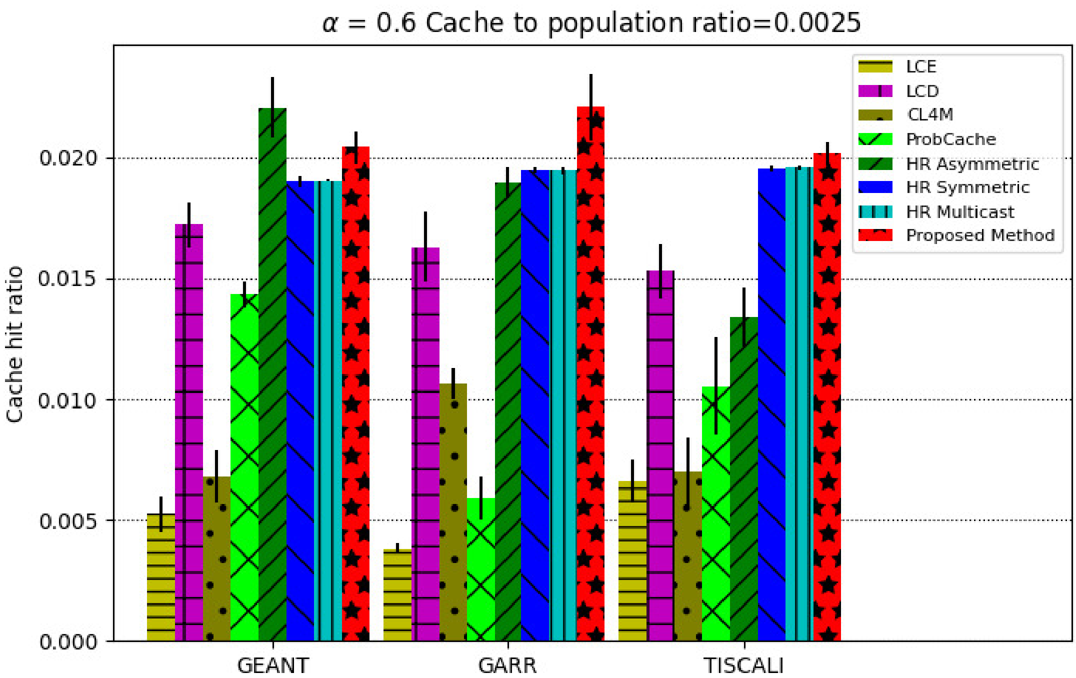


Figure 4.4: Cache hit vs Topologies, α=0.6 C=0.0025

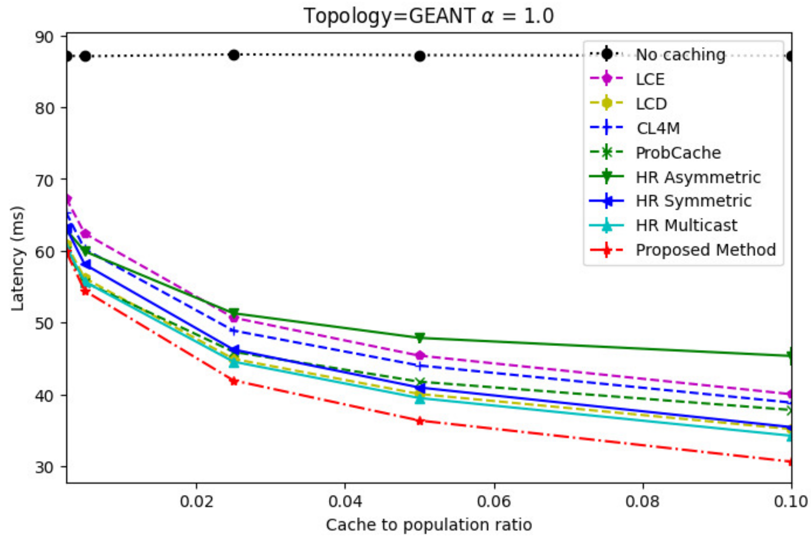
4.4.2 Impact on Latency

Though the cache hit ratio exhibits the percentage of the requested content server by the network cache, it can not provide the full picture of the schemes' performance. A consumer's experience greatly depends upon latency. Lower latency indicates better service of a network.

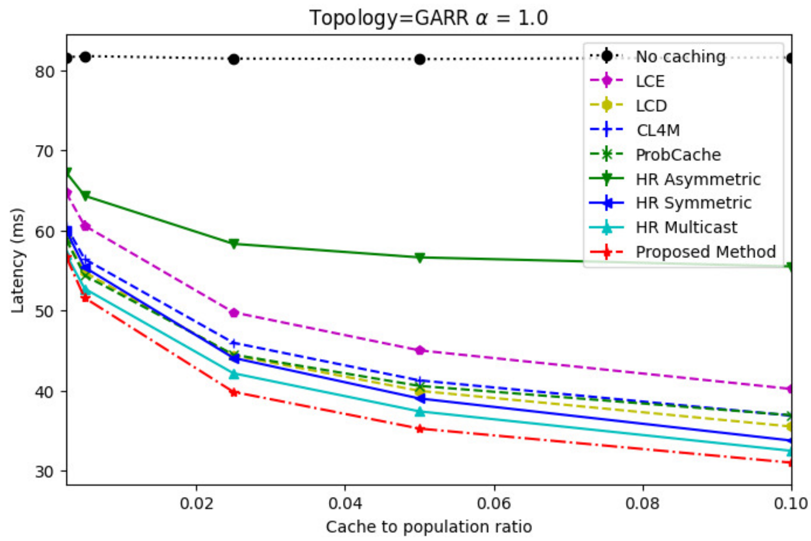
Figure 4.5 represents the performance of latency against varying cache sizes where skewness (α) is kept at 1.0. In the figure, latency is highest without any caching operation. In all topologies, it is observed that among caching schemes HR Asymmetric exhibits the highest latency whereas our proposed method shows the lowest latency in the network.

Similar performance is also observed when the content-to-population ratio is kept constant (C=0.025) in Fig. 4.6. The average entrance latency reduces as skewness becomes greater for all schemes. The latency of our proposed strategy is smaller than other schemes indicating better consumer service.

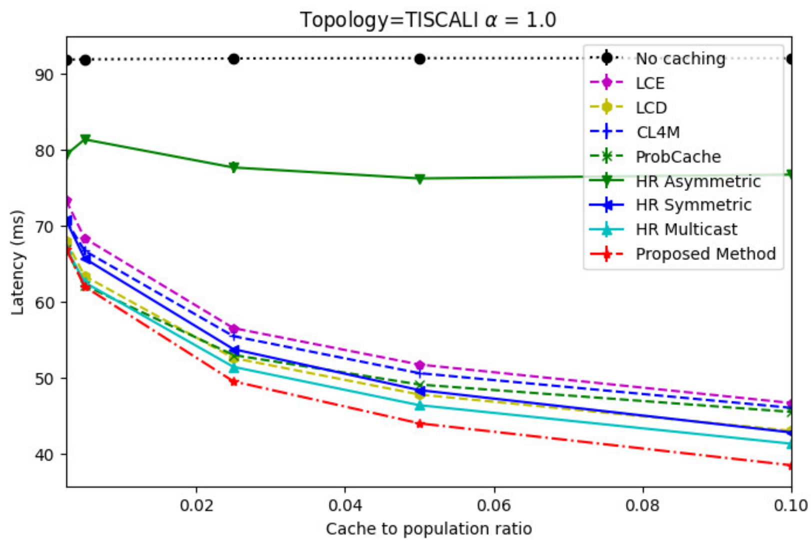
In Fig. 4.7, the average service latency of the strategies at (C=0.025, α =1.0) in three topologies is presented. This figure shows without any caching in the network shows the highest latency, in contrast, our designed strategy exhibits the lowest latency on all three topologies. As contents share local cache space in our scheme, it ensures popular content to be cached near to the consumer which also can provide satisfaction to the neighboring consumers to the greatest magnitude. As a result, in all three network topologies, the latency of the proposed strategy showed the best performance. On the other side, the "No caching" strategy exhibits the highest latency among all strategies as expected.



(a) GEANT, α=1.0

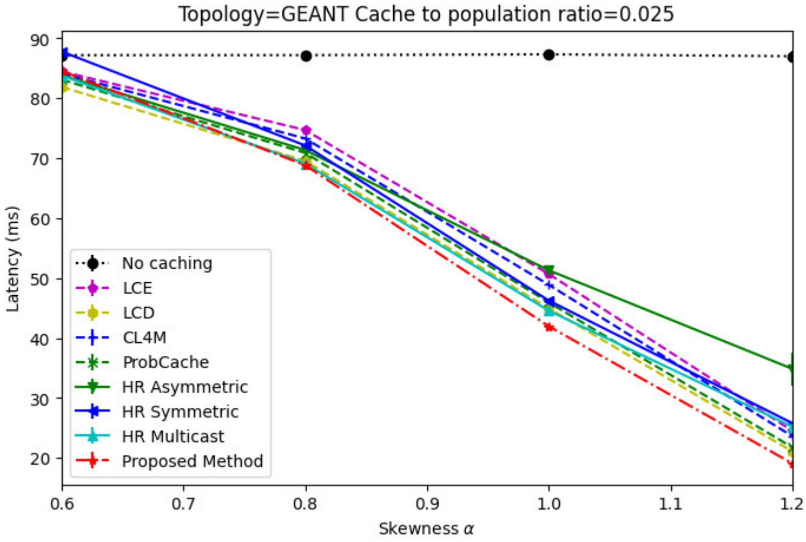


(b) GARR, α=1.0

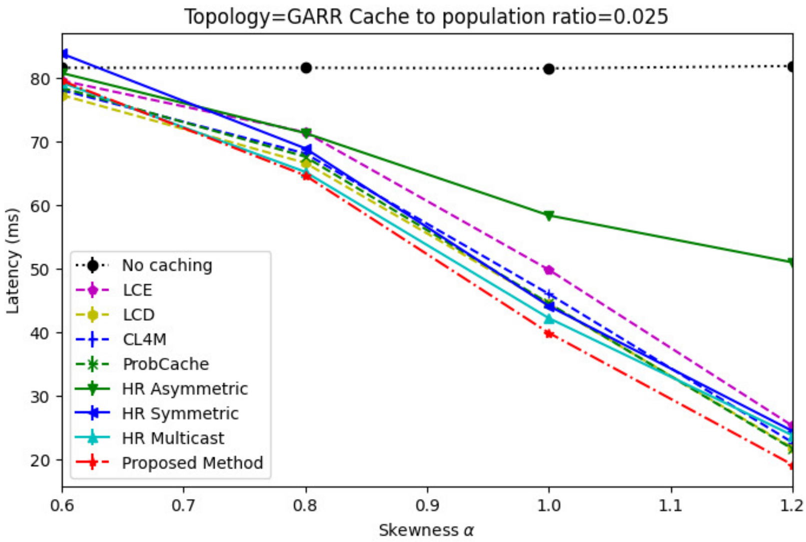


(c) TISCALI, α=1.0

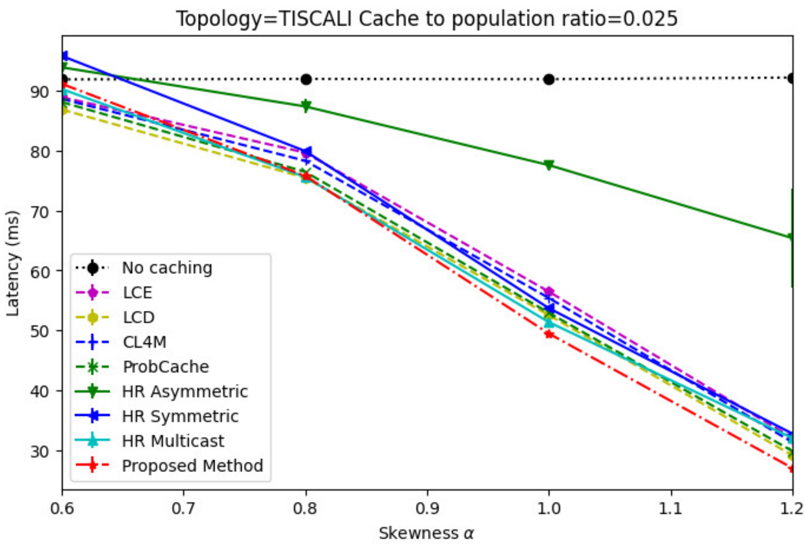
Figure 4.5: Performance of latency with different cache size ratio (C)



(a) GEANT, C=0.025



(b) GARR, C=0.025



(c) TISCALI, C=0.025

Figure 4.6: Performance of latency with different popularity skewness (α)

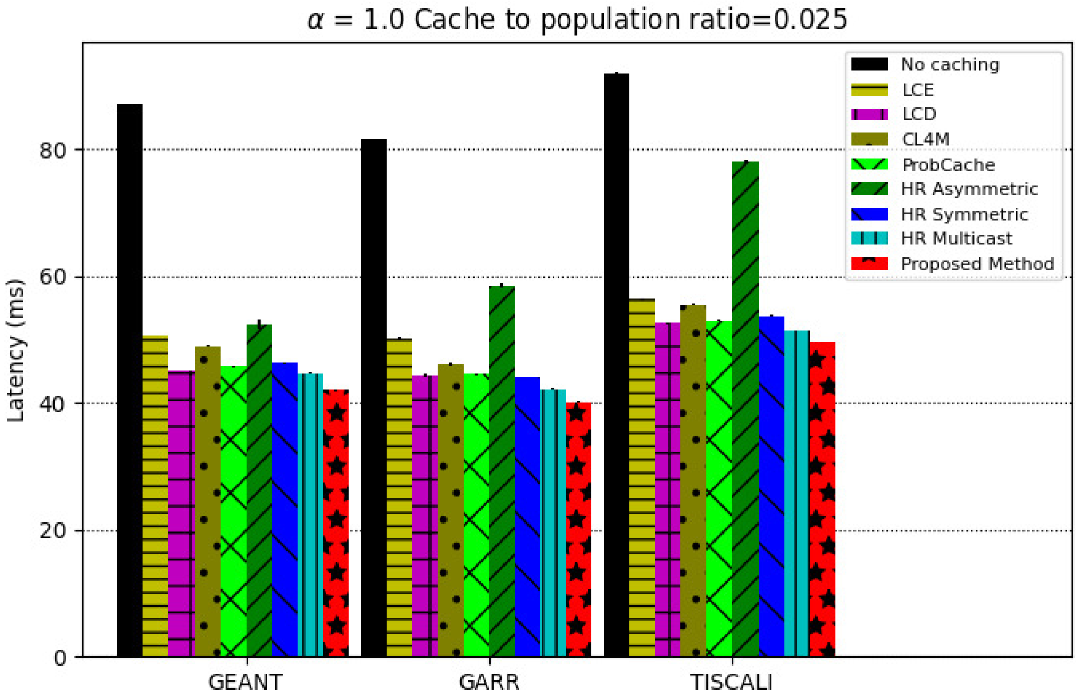


Figure 4.7: Latency vs Topologies, α=1.0 C=0.025

4.4.3 Impact on Link Load

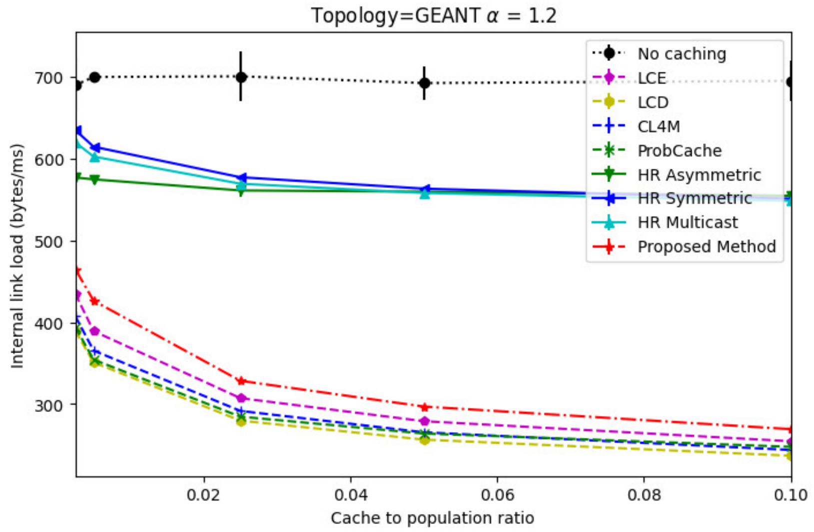
Link load is an important metric for examining network congestion. Heavy traffic is the main cause of network congestion which results in packet loss. It destroys the ability of a network to provide service to its consumers. When determining how much bandwidth is being used on a network link, one well-known performance statistic is link load. The mean traffic load of the network's links is referred to as the link load.

Fig. 4.8 demonstrates the link load results achieved in three topologies in terms of different cache-to-population ratios (C). In this segment of the investigation, skewness (α) is placed at 1.2. Although conventional hash routing schemes perform well regarding cache hit and latency, it's different in terms of link load. Because, in hash routing, sometimes it's necessary to detour the request packet and content packet out of the shortest path. From the figure, we can see similar observations. Conventional three hash routing schemes' link load is greater than other on-path caching schemes. Whereas our proposed scheme performs quite similarly to on-path caching in terms of link load. Moreover, on a large TISCALI network, it surpasses other schemes.

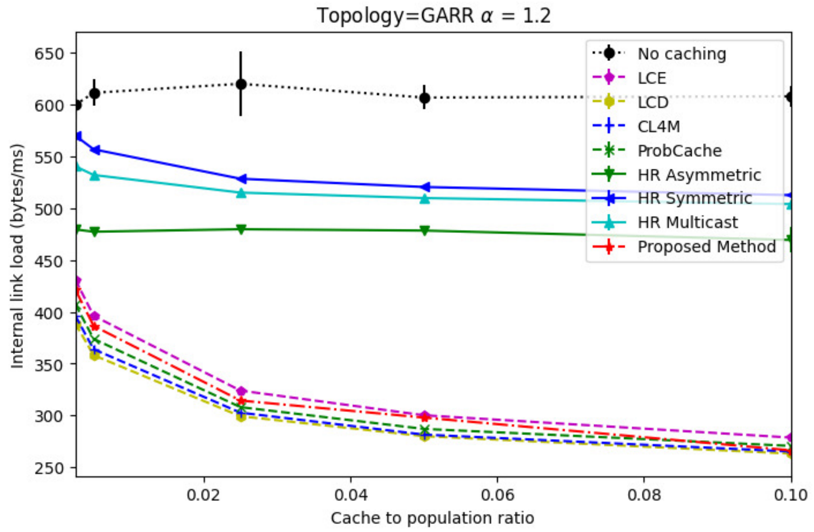
Similar performance is observed when the cache size is kept constant (C=0.1) varying popularity skewness α in Fig. 4.9. In both Fig. 4.8(c) and Fig. 4.9(c), the proposed strategy surpasses other caching schemes in the TISACALI network. In the case of GEANT and GARR topology, the link load of the proposed approach behaves quite similarly to on-path caching strategies. Link load greatly depends on the size of the network as well as how networks distribute their load among these nodes. The size of TISCALI is way greater than the other two networks. As the proposed approach, distributes its request load handling among all caching nodes, the benefit of the approach reflects more on the larger TISCALI network.

In Fig.4.10, the mean internal link load of the schemes at (C=0.1, α =1.2) in three topologies is presented. From the figure, it is clear that our proposed scheme shows a lower link load in all three scenarios which is quite similar to on-path caching.

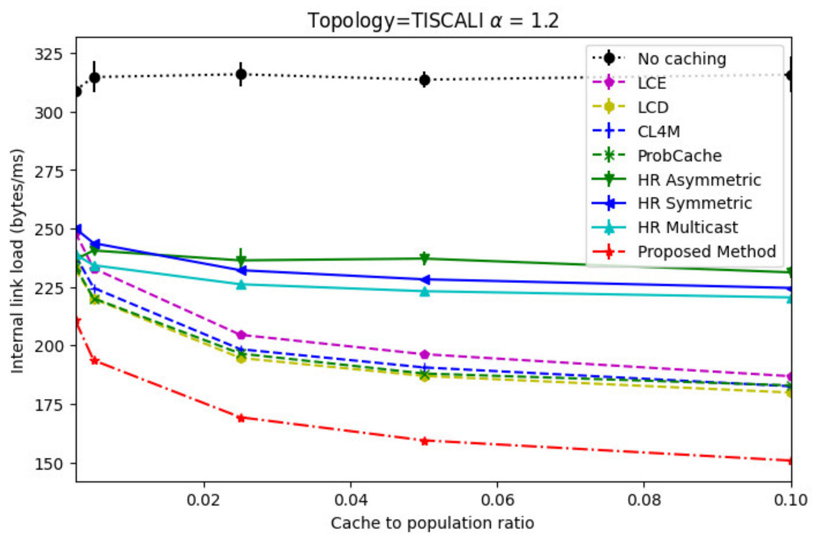
In this section, we rigorously investigated the performance of the proposed HCDF scheme in every possible scenario. Experiments were performed on both small and large network setups varying different skewness and cache to population ratio. From the findings, it's clear that the proposed HCDF scheme performed significantly better than existing well-known schemes.



(a) GEANT, α=1.2

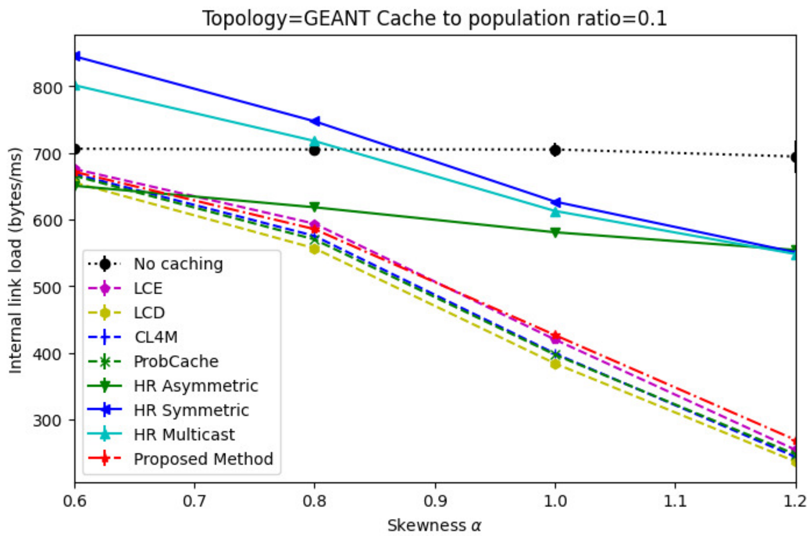


(b) GARR, α=1.2

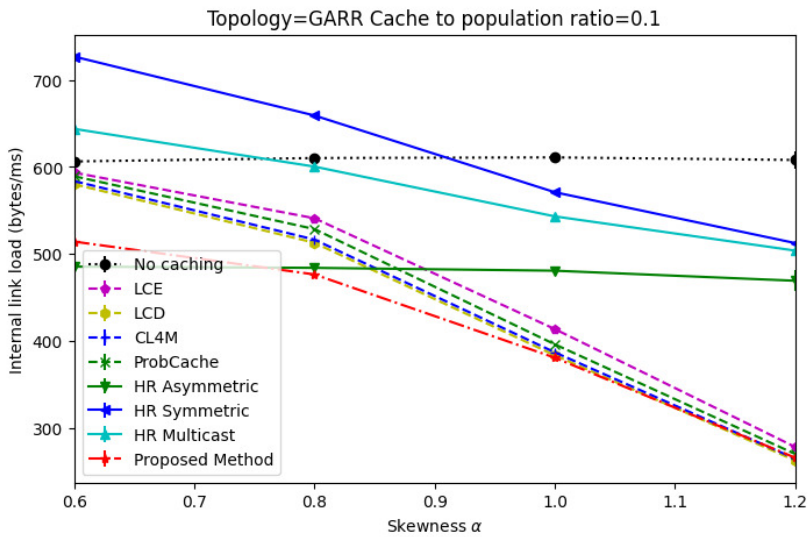


(c) TISCALI, α=1.2

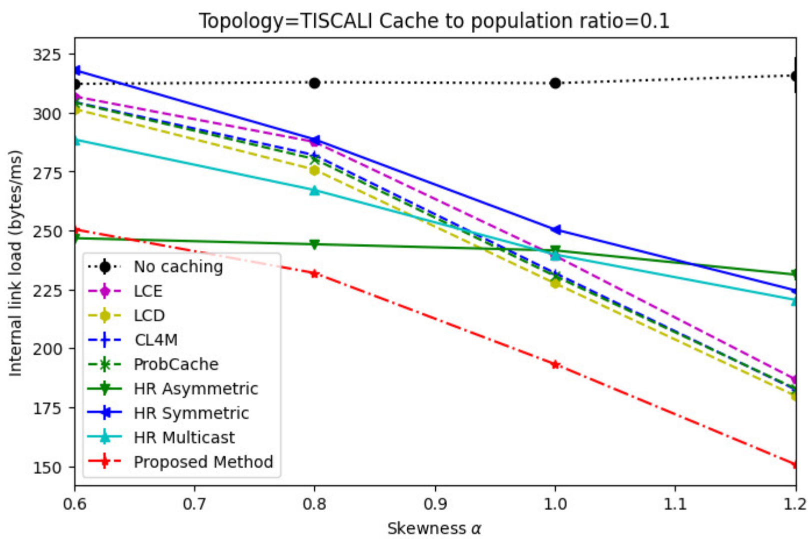
Figure 4.8: Link load performance with different cache size ratio (C)



(a) GEANT, C=0.1



(b) GARR, C=0.1



(c) TISCALI, C=0.1

Figure 4.9: Link load performance with different popularity skewness (α)

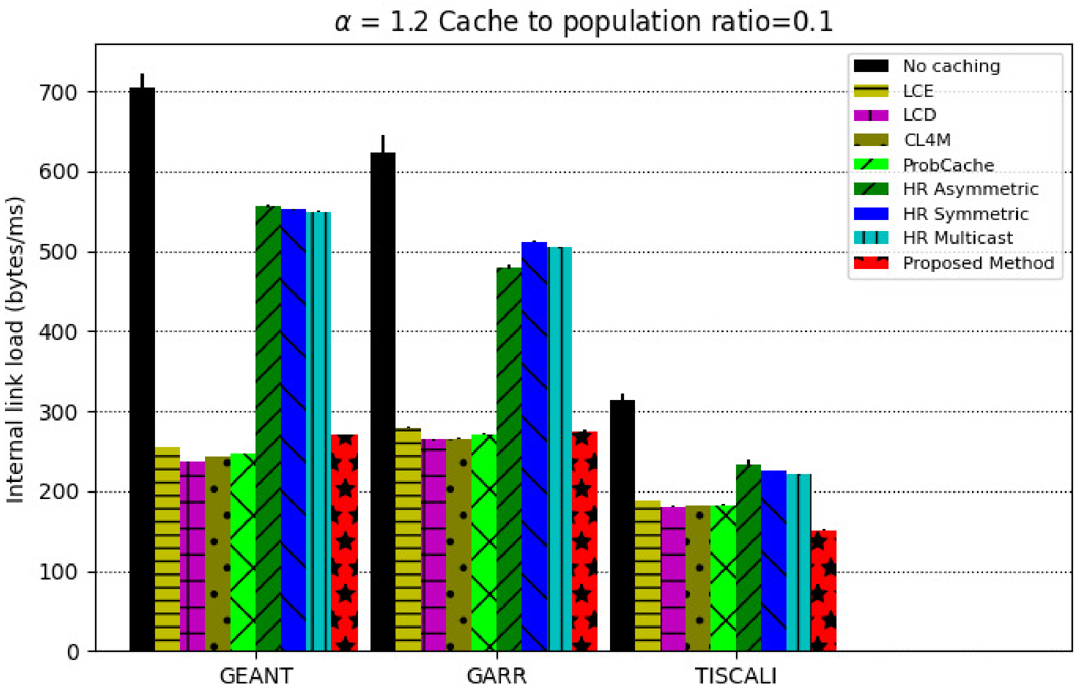


Figure 4.10: Link load vs Topologies, α=1.2 C=0.1

4.5 Qualitative Analysis

Qualitative analysis provides a complementary perspective to quantitative results by examining the conceptual and practical aspects of the proposed Hybrid Caching with Dynamic Forwarding (HCDF) strategy compared to on-path and off-path caching schemes. This analysis highlights the strengths, limitations, and use-case suitability of each approach to provide a holistic evaluation.

Table 4.2 and Table 4.3 present a qualitative comparison of the proposed HCDF strategy with traditional on-path and off-path caching schemes respectively.

Table 4.2: Qualitative Comparison of HCDF with Benchmark On-path Caching schemes

Aspect	**HCDF (Proposed)**	**LCE**	**LCD**	**CL4M**	**ProbCache**
Simplicity	Moderate; involves hybrid caching and dynamic forwarding strategies.	Simple but inefficient due to over-caching.	Very simple; caches only at downstream nodes.	Moderate complexity due to centrality calculations.	Moderately simple; uses probabilistic decisions.
Cache Efficiency	High; minimizes redundancy with dynamic content placement.	Low; excessive caching leads to high redundancy.	Moderate; limits caching to downstream nodes but lacks optimization.	High; efficiently caches content at central nodes.	Moderate; probabilistic caching balances efficiency.
Adaptability	High; adjusts caching and forwarding based on real-time traffic patterns.	None; fixed behavior, caches everywhere.	Low; caching is not influenced by network conditions.	Partial; depends on predefined centrality measures.	High; probabilistic caching adapts to traffic patterns.
Scalability	High; performs well in large, dynamic networks.	Poor; excessive caching strains network resources.	Low; limited to smaller or edge-focused networks.	Moderate; centrality computations pose scalability challenges.	High; well-suited for large, dynamic networks.
Latency Reduction	Significant; dynamic forwarding minimizes hop count and response time.	Minimal; redundancy does not guarantee optimized content delivery.	Moderate; reduces latency only at downstream nodes.	High; reduces latency for requests passing through central nodes.	Good; caches content closer to frequently accessed areas.
Deployment Ease	Moderate; requires setup for hybrid caching and dynamic forwarding policies.	Easy; straightforward to implement but resource-intensive.	Very easy; minimal configuration required.	Challenging; requires precomputation of centrality values.	Moderate; involves tuning probabilistic parameters.
Resource Utilization	High; efficiently uses cache space and reduces network congestion.	Poor; excessive resource usage due to indiscriminate caching.	Moderate; avoids redundancy but may underutilize cache space.	High; optimizes cache usage for central nodes.	Moderate; balances resource use through probabilistic caching.
Use-case Suitability	Best for dynamic, large-scale networks with varying traffic patterns.	Suitable for small-scale networks with no cache constraints.	Ideal for edge-focused or small-scale networks.	Effective in networks with identifiable central nodes.	Best for high-traffic, dynamic networks with diverse demands.

Table 4.3: Qualitative Comparison of HCDF with Off-path schemes

Aspect	**HCDF (Proposed)**	**HR Symmetric**	**HR Asymmetric**	**HR Multicast**
Simplicity	Moderate; involves hybrid caching and dynamic forwarding strategies.	Simple; symmetric paths with off-path caching.	Moderate complexity; requires managing asymmetric paths.	Moderate complexity due to multicast routing.
Cache Efficiency	High; reduces redundancy with dynamic content placement and forwarding.	Moderate; caches on alternative paths but lacks dynamic optimization.	High; avoids redundancy by using different forward and return paths.	Moderate; caches in multicast tree nodes but can lead to redundancy.
Adaptability	High; adjusts caching and forwarding based on traffic patterns.	Low; relies on predefined off-path strategies.	Moderate; adapts routing but not caching dynamically.	Moderate; partially adapts to multicast trees but not traffic dynamics.
Scalability	High; performs well in large, dynamic networks.	Moderate; scales reasonably well but may strain cache in larger networks.	High; asymmetric paths reduce congestion, suitable for larger networks.	Moderate; multicast caching is efficient but limited by tree size.
Latency Reduction	Significant; dynamic forwarding minimizes hop count and response time.	Moderate; reduces latency for off-path caching but not optimized.	High; asymmetric paths often reduce congestion and latency.	Moderate; multicast paths reduce redundant requests but not optimized for latency.
Deployment Ease	Moderate; requires configuration of hybrid caching and dynamic forwarding policies.	Easy; straightforward setup for symmetric off-path caching.	Moderate; requires implementation of asymmetric routing logic.	Challenging; multicast trees require complex configurations.
Resource Utilization	High; optimizes cache space and reduces network congestion.	Moderate; utilizes off-path nodes but can overuse resources.	High; asymmetric paths balance cache usage efficiently.	Moderate; caches in multicast trees but may waste resources in less-trafficked branches.
Use-case Suitability	Best for dynamic, large-scale networks with varying traffic patterns.	Suitable for networks with predictable off-path caching needs.	Effective for large-scale networks with heavy bidirectional traffic.	Suitable for multicast-heavy environments, such as video streaming.

The qualitative analysis demonstrates that the proposed HCDF strategy outperforms on-path and off-path caching schemes in terms of adaptability, scalability, and cache efficiency. While on-path caching is better suited for small or predictable networks and off-path caching excels in high-content diversity scenarios, HCDF provides a balanced and dynamic approach for modern, large-scale, and heterogeneous networks. These insights complement the quantitative evaluation, showcasing the practical viability and versatility of HCDF in diverse networking environments.

4.6 Trade-off

Trade-Off Between On-Path and Off-Path Caching In our proposed Hybrid Caching with Dynamic Forwarding (HCDF) strategy, both on-path and off-path caching strategies are utilized to leverage the strengths of each approach. However, this combination introduces certain trade-offs that must be considered, particularly regarding computational cost and network resource utilization.

1. On-Path Caching:

   - Advantages: On-path caching places content at intermediate nodes along the request path, ensuring that content can be served directly from nearby cache nodes, thus reducing latency and improving cache hit rates for subsequent requests.
   - Trade-off: The primary trade-off with on-path caching is the increased computational cost. Each router in the path must handle the additional responsibility of caching and updating content, leading to increased processing overhead. Furthermore, the strategy can result in redundant caching at multiple nodes along the same path, leading to inefficient use of cache space. This redundancy can cause increased network traffic, as content may be replicated across several nodes even when a cache hit could have been achieved with fewer copies.

2. Off-Path Caching:

   - Advantages: Off-path caching caches content at strategically selected nodes, often away from the main data path. This approach optimizes network resources and effectively uses cache space by avoiding redundant content replication at multiple intermediate nodes. Off-path caching typically results in lower network congestion and better scalability, especially in large-scale networks.
   - Trade-off: The main trade-off associated with off-path caching is higher latency. Since the content is cached at non-intermediate nodes, retrieving the cached data may require additional hops, resulting in longer retrieval times. Additionally, computational overhead is incurred in determining the most optimal caching locations and maintaining cache consistency across non-adjacent network segments. This process can involve more complex routing and state maintenance, which increases computational costs, particularly in dynamic environments with fluctuating content demand and network conditions.

3. Hybrid Caching (HCDF):

   By combining both on-path and off-path caching, the HCDF strategy aims to balance the benefits of both methods while minimizing their individual drawbacks. The system dynamically selects the most appropriate caching strategy based on network conditions, content popularity, and proximity to consumers. However, this hybrid approach introduces additional complexity in terms of computational cost for dynamically managing cache allocation and content placement across both on-path and off-path caches. Furthermore, managing the synchronization and consistency of content across both caching mechanisms adds a layer of complexity that must be handled efficiently to maintain performance.

In conclusion, while the on-path caching method offers the advantage of reduced latency and improved cache hit rates, it sacrifices network efficiency and increases computational overhead. On the other hand, off-path caching reduces redundancy and improves resource utilization but at the cost of increased latency and more complex management of cache consistency. The HCDF strategy mitigates these trade-offs by dynamically adjusting the caching strategy based on real-time network conditions and content demand, thus achieving an optimal balance between latency, computational cost, and resource utilization.

Chapter 5

Conclusion and Future Recommendations

In this chapter, we briefly summarized the major outcomes of this research and pointed out some future directions to work on. Finally, publications that are related to this thesis are listed.

In this research, we have designed an optimized caching policy named hybrid caching with dynamic forwarding (*HCDF*). Off-path caching schemes, for say standard hash routing schemes show higher cache hits with the disadvantage of detouring packets. To minimize this, we have proposed *HCDF*. In this scheme, local caching is facilitated in all cache nodes. To take advantage of this allocated space, we have designed an Estimation-based Content Placement (ECP) policy to bring popular content near to the consumer by caching them in shared cache space. The proposed HCDF scheme dynamically selects the best routing scheme among symmetric, asymmetric, and multicast based on hop count and path stretch. Extensive simulation has been performed on real-world topologies to evaluate our scheme. Simulation indicated that our proposed scheme reduces content retrieval latency, and link load over the network and increases cache hits compared to standard off-path and popular on-path caching strategies.

5.1 Limitations

The main limitations of the thesis are:

- Dependency on Network Topology: The network topology has a significant impact on the hybrid caching scheme's performance. Its effectiveness may differ in extremely irregular or limited networks, even though it has been tested on a variety of topologies.

- Fixed Cache Allocation Ratios: Cache space is divided between local and authoritative caches using preset ratios that might not adjust dynamically to traffic patterns or network conditions in real time.

- Scalability Issues for Extreme Scenarios: While the technique works well in large-scale configurations, networks with a high user density or content diversity may experience difficulties efficiently managing cache placements.

- Simulation-Driven Validation: A lot of simulations are used to evaluate the suggested plan, and actual execution may highlight other operational issues like hardware limitations or unexpected network dynamics.

5.2 Future Recommendations

In the future, we aim to study the effect of heterogeneous cache space allotment and partial content caching in our proposed scheme. Moreover, to address the limitations and improve the performance of the system, following future directions can be explored

- Dynamic Cache Allocation: Investigate mechanisms for adaptive cache allocation that can adjust the ratio of local to authoritative cache space in real-time based on network conditions and traffic patterns.

- Incorporation of Partial Caching: Explore the feasibility of partial content caching to enhance storage efficiency, particularly in networks with high content diversity.

- Heterogeneous Network Environments: Extend the evaluation of the hybrid caching scheme to include heterogeneous networks with varied hardware and software configurations to better understand its scalability and adaptability.

- Real-World Deployment and Testing: Transition from simulation-based validation to real-world implementation to identify operational challenges and refine the system further.

- Integration with Emerging Technologies: Explore how the hybrid caching strategy can integrate with emerging technologies like edge computing, 5G, and IoT for improved performance and broader applicability.

These recommendations aim to build on the strengths of the current work and address its limitations, paving the way for more versatile and practical implementations of hybrid caching schemes in diverse network scenarios.

5.3 List of Publications

The following publication is a direct consequence of the research carried out during the elaboration of the thesis and gives an idea of the progression that has been achieved.

1. **Dey, A.**, Asaduzzaman, & Arefin, M.S., "CyHy: A Centrality-aware Hybrid Caching Strategy in Information-centric Networking", *In Proceedings of the International Conference on Signal Processing, Information, Communication and Systems (IEEE SPICSCON)*, 2024.

2. **Dey, A.**, Asaduzzaman, Arefin, M.S., Dhar, P.K., & Shimamura, T., "A Hybrid Caching Scheme with Dynamic Forwarding Strategy in Information-Centric Network", in *IEEJ Transactions on Electrical and Electronic Engineering (TEEE), Volume 20, John Wiley & Sons*, 2024.

Bibliography

[1] S. Mastorakis, A. Afanasyev, Y. Yu, and L. Zhang, "ntorrent: Peer-to-peer file sharing in named data networking," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–10, IEEE, 2017.

[2] C. V. Priscilla and A. Charulatha, "A comparative study on caching strategies in content centric networking for mobile networks," in *2019 11th International Conference on Advanced Computing (ICoAC)*, pp. 122–128, IEEE, 2019.

[3] 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 *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 181–192, 2007.

[4] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, "Network of information (netinf)–an information-centric networking architecture," *Computer Communications*, vol. 36, no. 7, pp. 721–735, 2013.

[5] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[6] O. Khalid, I. A. Khan, R. N. B. Rais, and A. Abbas, "Caching, security, and mobility in content-centric networking," *Fog Computing: Theory and Practice*, pp. 79–103, 2020.

[7] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. L. Aguiar, and A. V. Vasilakos, "Information-centric networking for the internet of things: challenges and opportunities," *IEEE Network*, vol. 30, no. 2, pp. 92–100, 2016.

[8] N. Dutta, H. K. D. Sarma, R. Jadeja, K. Delvadia, and G. Ghinea, "Introduction to information-centric networks," in *Practical Networking*, pp. 1–25, Springer International Publishing, 2021.

[9] Q. N. Nguyen, J. López, T. Tsuda, T. Sato, K. Nguyen, M. Ariffuzzaman, C. Safitri, and N. H. Thanh, "Adaptive caching for beneficial content distribution in information-centric networking," in *2020 International Conference on Information Networking (ICOIN)*, pp. 535–540, IEEE, 2020.

[10] I. U. Din, S. Hassan, M. K. Khan, M. Guizani, O. Ghazali, and A. Habbal, "Caching in information-centric networking: Strategies, challenges, and future research directions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1443–1474, 2017.

[11] F. Yan, H. Luo, S. Zhang, Z. Wang, and P. Lian, "A comparative study of ip-based and icn-based link-state routing protocols in leo satellite networks," *Peer-to-Peer Networking and Applications*, vol. 16, p. 3032–3046, Sept. 2023.

[12] M. Aldaoud, D. Al-Abri, M. Awadalla, and F. Kausar, "Leveraging icn and sdn for future internet architecture: A survey," *Electronics*, vol. 12, p. 1723, Apr. 2023.

[13] "Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper — cisco.com." https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html. [Accessed 11-01-2025].

[14] "Mobile data traffic forecast – Ericsson Mobility Report." https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-traffic-forecast. [Accessed 17-07-2024].

[15] Research and Markets, "Global Mobile Data Traffic Strategic Analysis Report 2023-2030." https://www.researchandmarkets.com/report/mobile-data-traffic. [Accessed 17-07-2024].

[16] M. I. A. Zahed, I. Ahmad, D. Habibi, Q. V. Phung, M. M. Mowla, and M. Waqas, "A review on green caching strategies for next generation communication networks," *IEEE Access*, vol. 8, pp. 212709–212737, 2020.

[17] Y. Gui and Y. Chen, "A cache placement strategy based on compound popularity in named data networking," *IEEE Access*, vol. 8, pp. 196002–196012, 2020.

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

[19] 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.

[20] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *Proceedings of the second edition of the ICN workshop on Information-centric networking*, pp. 55–60, 2012.

[21] I. Psaras, W. K. Chai, and G. Pavlou, "In-network cache management and resource allocation for information-centric networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 11, pp. 2920–2931, 2013.

[22] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache "less for more" in information-centric networks (extended version)," *Computer Communications*, vol. 36, no. 7, pp. 758–770, 2013.

[23] K. N. Lal and A. Kumar, "A centrality-measures based caching scheme for content-centric networking (ccn)," *Multimedia Tools and Applications*, vol. 77, no. 14, pp. 17625–17642, 2018.

[24] J. Pfender, A. Valera, and W. K. Seah, "Easy as abc: A lightweight centrality-based caching strategy for information-centric iot," in *Proceedings of the 6th ACM conference on information-centric networking*, pp. 100–111, 2019.

[25] F. Qazi, O. Khalid, R. N. B. Rais, I. A. Khan, *et al.*, "Optimal content caching in content-centric networks," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.

[26] Y. Li, J. Wang, and R. Han, "An on-path caching scheme based on the expected number of copies in information-centric networks," *Electronics*, vol. 9, no. 10, p. 1705, 2020.

[27] Q. N. Nguyen, J. Liu, Z. Pan, I. Benkacem, T. Tsuda, T. Taleb, S. Shimamoto, and T. Sato, "Ppcs: A progressive popularity-aware caching scheme for edge-based cache redundancy avoidance in information-centric networks," *Sensors*, vol. 19, no. 3, p. 694, 2019.

[28] Y. Li, J. Wang, and R. Han, "PB-NCC: A popularity-based caching strategy with number-of-copies control in information-centric networks," *Applied Sciences*, vol. 12, p. 653, Jan. 2022.

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

[30] V. Sourlas, I. Psaras, L. Saino, and G. Pavlou, "Efficient hash-routing and domain clustering techniques for information-centric networks," *Computer Networks*, vol. 103, pp. 67–83, 2016.

[31] K. Hasan and S.-H. Jeong, "Efficient caching for delivery of multimedia information with low latency in icn," in *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 745–747, IEEE, 2019.

[32] H. Jung and S. Kim, "A multiple hash routing scheme for fast data retrieval in icn," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1562–1565, 2018.

[33] A. Azgin, R. Ravindran, and G. Wang, "Hash-based overlay routing architecture for information centric networks," in *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–9, 2016.

[34] K. Thar, N. H. Tran, S. Ullah, T. Z. Oo, and C. S. Hong, "Online caching and cooperative forwarding in information centric networking," *IEEE Access*, vol. 6, pp. 59679–59694, 2018.

[35] N. Kamiyama and M. Murata, "Dispersing content over networks in information-centric networking," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 521–534, 2019.

[36] H. Wu, J. Li, J. Zhi, Y. Ren, and L. Li, "A hybrid icn caching strategy based on region division," in *Proceedings of the 15th International Conference on emerging Networking EXperiments and Technologies*, pp. 78–79, 2019.

[37] A. Kalla and S. Sharma, "Exploring off-path caching with edge caching in information centric networking," in *2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT)*, pp. 630–635, IEEE, 2016.

[38] M. A. Naeem, M. A. U. Rehman, R. Ullah, and B.-S. Kim, "A comparative performance analysis of popularity-based caching strategies in named data networking," *IEEE Access*, vol. 8, pp. 50057–50077, 2020.

[39] L. Saino, I. Psaras, and G. Pavlou, "Icarus: A caching simulator for information centric networking (icn)," in *Proceedings of the 7th International ICST Conference on Simulation Tools and Techniques*, SIMUTools '14, (Brussels, BEL), p. 66–75, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

[40] "The internet topology zoo." http://www.topology-zoo.org/dataset.html. (Accessed on 21 May 2024).

[41] "Rocketfuel maps." https://research.cs.washington.edu/networking/rocketfuel/. (Accessed on 21 May 2024).

Ashim Dey

ORIGINALITY REPORT

16%	11%	12%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.researchgate.net Internet Source	1%
2	Submitted to Chittagong University of Engineering and Technology Student Paper	1%
3	www.mdpi.com Internet Source	1%
4	etd.uum.edu.my Internet Source	1%
5	lib.buet.ac.bd:8080 Internet Source	1%
6	dokumen.pub Internet Source	<1%
7	www.grin.com Internet Source	<1%
8	Lorenzo Saino, Ioannis Psaras, George Pavlou. "Hash-routing schemes for information centric networking", Proceedings of the 3rd ACM SIGCOMM workshop on	<1%