Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 9, No. 2, April, 2025

# CacheCraft: A Topology-Aware PageRank Centrality Algorithm for Cache Optimization in Named Data Networking

Ridha M. Negara [1, 2, 3*], Nana R. Syambas [1], Eueung Mulyana [1], Rashid M. Fajri [2], Mochamad S. Budiana [1, 2]

[1] School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung, Indonesia.

[2] School of Electrical Engineering, Telkom University, Bandung, Indonesia.

[3] The University Center of Excellence for Intelligent Sensing-IoT, Telkom University, Bandung, Indonesia.

**Abstract**

This study introduces CacheCraft, a novel approach for heterogeneous Content Store (CS) capacity allocation in Named Data Networking (NDN). Traditional NDN allocates CS capacity uniformly across routers, assuming equal storage requirements for all nodes. However, user content preferences and traffic patterns vary significantly, necessitating a more tailored allocation strategy. Additionally, the complexity of network topologies exacerbates the challenge, as static and homogeneous CS allocations lead to inefficiencies, increased latency, and reduced cache effectiveness in dynamic and dense networks. CacheCraft addresses these challenges by leveraging the PageRank algorithm to calculate the centrality of each node in the network. This centrality value determines the proportion of CS capacity assigned to each node, optimizing storage for nodes with higher traffic and strategic importance. The use of PageRank ensures scalable and reliable centrality computation, even in complex topologies. The performance of CacheCraft is validated across diverse network scenarios, including topologies of varying complexity, using metrics such as Cache Hit Ratio (CHR), average latency, and time complexity. Experimental results demonstrate that CacheCraft achieves an average improvement of 7.8% in CHR and a 5.6 ms reduction in latency compared to state-of-the-art methods. Moreover, CacheCraft maintains algorithmic computational efficiency, making it suitable for real-world deployment in complex and dynamic NDN environments. These findings highlight CacheCraft as a robust and scalable solution for optimizing NDN performance through adaptive and efficient CS capacity allocation.

**Keywords:**

Named Data Networking;
Caching Placement Strategy;
PageRank Centrality;
Topology-Aware Caching;
Heterogeneous Content Store Allocation;
In-Network Caching.

**Article History:**

| | | | |
|---|---|---|---|
| **Received:** | 03 | December | 2024 |
| **Revised:** | 11 | February | 2025 |
| **Accepted:** | 18 | February | 2025 |
| **Published:** | 01 | April | 2025 |

## 1- Introduction

The rapid growth of internet users has compelled service providers to anticipate increasing demands to ensure reliable quality of service (QoS) [1, 2]. As more devices connect to the network, the volume of data requested and transmitted grows exponentially [3, 4]. Traditional internet architectures, being host-centric, are inadequate for managing the complexity of future networks. These architectures rely heavily on location-based data retrieval, creating significant bottlenecks and increasing latency when multiple users request identical content [5]. This limitation poses a considerable challenge in maintaining the quality of experience (QoE) for users in high-demand environments [6, 7].

To address these challenges, Named Data Networking (NDN) has emerged as a promising paradigm for the future internet. Unlike traditional approaches, NDN adopts a content-centric or information-centric methodology, which is

* **CONTACT**: ridhanegara@telkomuniversity.ac.id

**DOI:** http://dx.doi.org/10.28991/ESJ-2025-09-02-09

particularly well-suited for environments such as the Internet of Things (IoT), multimedia content delivery, and ad hoc networks [3, 5]. Its in-network caching mechanism is a game changer, as it reduces latency, improves QoE, and lowers operational costs. By caching previously accessed content at NDN routers, NDN minimizes dependency on original data sources, enabling faster content delivery [8]. For example, multimedia applications in NDNs can deliver cached content with reduced buffering and quicker load times, meeting the growing demand for high-quality streaming and real-time services [9, 10].

A fundamental challenge in NDN is determining "WHERE" to store content and "WHAT" content to cache [11]. Accurate caching decisions directly impact network performance, especially in large-scale networks with complex and dynamic topologies [12, 13]. While caching replacement policies, such as least recently used (LRU) policies, influence content eviction processes, caching placement strategies (CPSs) have a more significant effect on enhancing network performance [14, 15]. Therefore, optimizing the CPS in the NDN, with an emphasis on topology awareness, is crucial.

The effectiveness of a caching placement strategy is heavily influenced by the underlying network topology [16]. On-path caching strategies, such as Content Cache Everywhere (CEE), store content along the path from the data source to the user [15]. While effective in static networks, these strategies become less efficient in dynamic topologies, such as those found in IoT or mobile networks, where access paths frequently change [17]. Cached content on outdated paths becomes underutilized, reducing overall system efficiency. Conversely, off-path caching strategies store content at alternative nodes, offering greater adaptability but at the cost of greater complexity because of the need for optimal location analysis [18, 19].

Topology-aware caching strategies provide promising solutions by leveraging network topology characteristics to identify strategically essential nodes [20, 21]. Centrality metrics, such as betweenness centrality and closeness centrality, can effectively prioritize nodes for caching on the basis of their connectivity and role within the network [1]. However, these metrics are often computationally expensive, particularly in large-scale and dynamic networks [6, 22, 23]. In contrast, PageRank centrality offers a scalable and computationally efficient alternative, balancing node importance with network connectivity [24, 25]. For example, in networks such as the Palapa Ring, PageRank centrality has been shown to increase caching efficiency while reducing computational overhead, making it highly suitable for real-world implementations.

Even though existing centrality-based caching strategies, such as PBCE [26] and DCAM [27], have made progress in optimizing cache placement and performance metrics, they often assume homogeneous cache size allocation, which fails to address the varying traffic demands of real-world networks. On the other hand, methods such as C3CPS [28] and betweenness-centrality-based approaches [29] introduce significant computational overhead, making them impractical for large-scale or dynamic topologies. These challenges highlight the need for an adaptive, scalable caching framework capable of dynamically allocating cache sizes on the basis of real-time network dynamics—a gap that this study seeks to fill.

This study proposes CacheCraft, a topology-aware caching framework that leverages PageRank centrality to optimize content store (CS) capacity allocation. By dynamically allocating heterogeneous CS capacities and strategically caching popular content, CacheCraft ensures efficient resource utilization and improved network performance. Unlike existing approaches, CacheCraft integrates topology awareness into caching decisions, prioritizing nodes on the basis of both connectivity and traffic dynamics. The key contributions of this research include the following:

- A systematic approach that integrates placement, size dimensioning, and replacement processes is developed to optimize content storage and enhance NDN performance.
- An adaptive caching strategy that leverages PageRank centrality for efficient and scalable allocation of CS capacity on the basis of node significance is introduced.
- Improving centrality metrics to address the challenges posed by large-scale and dynamic topologies, ensures enhanced adaptability and performance.
- Validating CacheCraft's effectiveness via real-world topologies, including the GARR, Palapa Ring, and Tiscali.
- Conducting comprehensive simulations on the ICARUS platform to evaluate the impact of content popularity, topology complexity, and cache capacity, benchmarking CacheCraft against state-of-the-art techniques.

The remainder of this paper is organized as follows: Section 2 provides an overview of the theoretical foundation, discusses existing caching mechanisms, and highlights the limitations of current approaches, culminating in a comparative analysis and the introduction of CacheCraft. Section 3 presents a detailed explanation of the proposed topology-aware dynamic caching allocation mechanism, including parameter definitions and quota computation methods. Section 4 outlines the experimental setup, evaluates the performance through various metrics, and discusses the results. Finally, Section 5 concludes the study and suggests directions for future work.

## 2- Literature Review

### 2-1- Basic Theory and Formulas

The centrality value serves as a reference for determining the cache capacity size of each node. Nodes with higher centrality values are assigned to larger cache capacities to ensure optimal resource utilization. This section outlines commonly used centrality algorithms that form the foundation of network analysis [30, 31]:

#### a. Uniform Centrality

Cache capacities are distributed evenly across all nodes by dividing the total cache capacity by the number of available nodes, assuming uniformity in network importance.

#### b. Degree Centrality

Degree centrality ranks nodes on the basis of their connectivity. Nodes with more direct connections are assigned proportionally larger cache capacities [32].

#### c. Betweenness Centrality

Betweenness centrality measures the extent to which a node lies on the shortest path between other nodes [33]:

$$BC\ (i) = \sum_{s,t\,\in\,i} \frac{\sigma(s,t|i)}{\sigma(s,t)} \tag{1}$$

where $\sigma\ (s,t)$ is the number of shortest paths between nodes $s$ and $t$, and $\sigma(s,t|\mathrm{i})$ is the number passing through node i. Formula 1 is used to determine the centrality value of cache nodes in the topology. This value is then used to allocate the size of the cache capacity proportionally.

#### d. Eigenvector Centrality

The centrality value of a node depends not only on the number of neighbouring nodes, but also on its centrality value. Equation 2 is used to calculate the centrality of eigenvector [30].

$$EC(i) = \frac{1}{\lambda}\sum_{j=1}^{n} A_{ij} EC(j) \tag{2}$$

Let $EC(i)$ be the eigenvector centrality (EC) for node i, $A_{ij}$ be the connection between node i and node j, λ be the largest eigenvalue of A, $EC(j)$ be the EC of node j and n is the total number of nodes. To calculate EC, it is necessary to find the value of the largest eigenvector λ and the associated vector of eigenvectors. With simple rearrangement we can express it as an eigenvector equation in Equation 3 by defining the vector $\overrightarrow{EC} = (EC(1), EC(2), EC(3) \dots)$.

$$A\overrightarrow{EC} = \lambda\overrightarrow{EC} \tag{3}$$

#### e. Closeness Centrality

Closeness centrality quantifies the proximity of a node to all other nodes in the network. The node's ability to disseminate information throughout the topology is demonstrated [34, 35]. Equation 4 presents the mathematical expression for closeness centrality. In this equation, n represents the total number of nodes in the network, CC(i) is the closeness centrality at node i, and d(i, j) represents the shortest distance between nodes i and j [29].

$$CC(i) = \frac{n-1}{\sum_{j=1, j\neq i}^{n} d(i,j)} \tag{4}$$

#### f. PageRank Centrality

PageRank centrality, a variant of eigenvector centrality developed by Google [36, 37], ranks nodes on the basis of their connectivity and influence within the network. Unlike other centrality measures, it incorporates both direct and indirect connections, making it well- suited for dynamic network environments. The mathematical expression is [38]:

$$PR_k(i) = \begin{cases} \frac{1}{n}\ , & k = 1 \\ \frac{(1-d)}{n} + d\ x\ \sum_{j\,\in\,R^N} \frac{PR_{k-1}(j)}{L(j)}, & k > 1 \end{cases} \tag{5}$$

k = {1, 2, 3, - .., kmax}

where $k$ is the iteration number, $i$ is the current node, $PR_k(i)$ is the PageRank centrality at iteration $k$, $n$ is the number of nodes, $RN$ is the set of neighbors of node $i$, $L(j)$ is the number of links of node $j$, and $d$ is the damping factor (ranging from 0 to 1). $d$ represents the probability of the user accessing new content randomly. The closer d is to 1, the greater the probability of the user accessing new content randomly. This mechanism allows reliable centrality calculations by predicting user behavior across the network.

By integrating traffic flow and topological influence, PageRank centrality addresses the limitations of traditional centrality metrics, ensuring that caching placement decisions are both scalable and efficient.

### *2-2- Overview of Named Data Networking (NDN)*

Named Data Networking (NDN) represents a transformative shift from traditional host-centric internet architectures to content-centric networking, where content is accessed by its name rather than its location [11, 39]. One of NDN's core innovations is in-network caching, which allows routers to temporarily store content along the delivery path. This mechanism reduces latency, enhances the quality of experience (QoE), and optimizes bandwidth usage [11, 40].

However, the default caching strategy in NDN, known as Cache Everything Everywhere (CEE) [41], while simple, suffers from significant drawbacks. CEE replicates content indiscriminately across all routers, leading to redundant content storage, inefficient use of cache resources, and increased bandwidth consumption. These limitations highlight the need for more sophisticated caching mechanisms that optimize cache placement and resource allocation while minimizing redundancy.

Figure 1 illustrates the broader in-network caching mechanisms, comprising four interconnected processes: cache placement, cache size dimensioning, content placement, and cache replacement. These processes encompass diverse strategies to address bandwidth consumption, resource allocation, and content eviction.

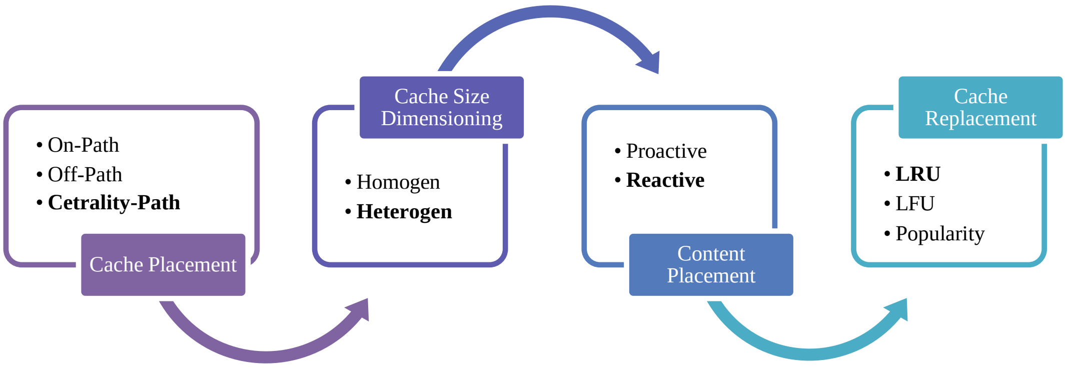



**Figure 1. Overview of In-Network Caching Mechanisms and CacheCraft's Selected Approaches**

The proposed framework, CacheCraft, selectively employs specific mechanisms from each process to optimize caching performance in dynamic and complex network environments:

- Cache placement: Implements a topology-aware centrality- path strategy, prioritizing nodes on the basis of their centrality within the network topology to increase efficiency.
- Cache size dimensioning: This method adopts heterogeneous allocation, aligning the cache capacity with node significance to address traffic variability.
- Content placement: This uses a reactive approach, dynamically adjusting content placement in response to demand patterns.
- Cache replacement: This method relies on the LRU policy to manage content eviction efficiently, ensuring high cache hit ratios.

While Figure 1 highlights the full spectrum of in-network caching strategies, CacheCraft's targeted selection demonstrates its suitability for real-world deployment, balancing computational efficiency with improved network performance.

### *2-3- Cache Placement Strategies*

#### *2-3-1- On-Path and Off-Path Caching*

Traditional caching placement strategies are categorized as on-path and off-path caching. On-path caching, as implemented in the content cache everywhere (CEE) approach, stores content along the delivery path. While this ensures fast access for repeat requests, it suffers from inefficiency in dynamic topologies, such as mobile or IoT networks, where frequent path changes render cached data inaccessible. Conversely, off-path caching distributes content across alternative nodes, optimizing cache loads but at the cost of increased computational complexity in determining optimal locations [14, 20, 28, 42, 43].

CacheCraft addresses these limitations through a topology-aware caching strategy, which combines the adaptability of off-path caching with the efficiency of dynamic placement. In real-world simulations, CacheCraft reduced the average latency by 6.8 ms compared with that of betweenness-based off-path strategies, demonstrating its ability to optimize performance even in dynamic topologies such as the Palapa Ring.

### *2-3-2- Centrality-Based Caching Placement*

Centrality-based approaches have emerged as powerful tools for optimizing caching placement by leveraging network topology characteristics. Betweenness centrality, as demonstrated by Lal & Kumar [29], and He et al. [44], prioritizes nodes that lie on the shortest paths between others. While this approach improves CHR by up to 10% over uniform placement in smaller networks, its computational cost grows exponentially with network size, as seen in the Tiscali topology. Closeness centrality focuses on nodes with minimal cumulative distances to others, reducing latency by placing caches closer to end-users. Studies by Amadeo et al. [45], and Koide et al. [46] demonstrated latency reductions of up to 7 ms compared with uniform cache placement. However, their static computation of centrality values limits adaptability in dynamic environments.

### *2-4- Cache Size Dimensioning (CSD)*

Cache size dimensioning (CSD) determines the optimal capacity of Content Stores (CS) at each node. Traditional methods often assume homogeneous cache sizes, as seen in studies by Rossi & Rossini [47], and Lal & Kumar [29]. While this simplifies implementation, it fails to account for variations in traffic demand, leading to inefficiencies in cache utilization. Uniform allocation resulted in up to a 30% increase in cache miss rates in GARR topologies due to congestion at high-traffic nodes.

CacheCraft addresses these issues by employing heterogeneous CS allocation on the basis of PageRank centrality. Nodes with higher centrality values are allocated larger cache capacities, ensuring efficient storage of frequently accessed content. The simulation results show that CacheCraft achieves an average latency reduction of 5.6 ms compared to homogeneous strategies, making it particularly effective for diverse and large-scale network scenarios.

### *2-5- Limitations of the Current Approach*

Despite advancements in caching strategies, several limitations persist:

#### *1. Redundancy in Centrality-Based Methods*

Centrality-based schemes such as CMBA [29] suffer from redundancy when nodes with identical centrality values store the same content, leading to inefficient resource use. CacheCraft eliminates this redundancy by incorporating PageRank's ability to prioritize nodes on the basis of connectivity and influence, reducing content duplication by 15% compared with CMBA.

#### *2. Static Assumptions in Topology*

Strategies such as NICE [48] perform well in static networks but struggle with dynamic environments such as the IoT or mobile networks. CacheCraft's dynamic adaptability allows it to achieve higher CHR and lower latency in scenarios with frequent topology changes, such as the Palapa ring network.

#### *3. Computational Complexity*

Betweenness and closeness centralities incur exponential computation times in large networks. For example, betweenness-based methods experience 50% greater processing times in Tiscali than CacheCraft does, which maintains efficiency through iterative PageRank calculations.

### *2-5-1- Importances of Centrality Paths and Cache Size Dimensioning*

Centrality-based caching strategies have been widely adopted in networking paradigms, especially for their ability to prioritize strategically important nodes in cache placement. These methods leverage metrics such as betweenness, closeness, and custom centrality measures to optimize content delivery and reduce latency. Table 1 provides a detailed review of centrality-path-based caching strategies, focusing on their placement techniques, cache size dimensioning methods, and centrality approaches.

**Table 1. Review of Centrality-Based Caching Strategies and Cache Size Dimensioning in Networking Paradigms**

| References | Name Scheme | Cache Placement | Cache Size Dimensioning | Centrality Method | Network Scenario | Key Observation |
|---|---|---|---|---|---|---|
| Luo & An (2017) [49] | NCBIC | Centrality-path | Homogeneous | Betweenness | CCN | Balances caching load among nodes but faces cache overload on central nodes. |
| An & Luo (2018) [50] | EEBINC | Off-path | Heterogeneous | Betweenness | CDN | Focuses on minimizing energy consumption but suffers from high computational complexity. |
| Lal & Kumar (2018) [29] | CMBA | Centrality-path | Homogeneous | Betweenness | CCN | Effective in static topologies but has limited adaptability to dynamic networks. |
| Khan et al. (2018) [48] | NICE | Centrality-path | Homogeneous | Custom | ICN | Enhances eviction rates but struggles to adapt to frequent topology changes in dynamic networks. |
| Lal & Kumar (2019) [26] | PBCE | Centrality-path | Homogeneous | Betweenness | NDN | Improves cache performance but creates redundancy when nodes share identical centrality values. |
| Zheng et al. (2019) [51] | BEP | Centrality-path | Heterogeneous | Betweenness | CCN | Uses edge popularity but experiences high node replacement rates, reducing cache stability. |
| Delvadia et al. (2019) [52] | ERS | Centrality-path | Homogeneous | Betweenness | ICN | Optimizes routing but assumes uniform cache sizes, limiting scalability in large networks. |
| Meng et al. (2019) [53] | DCS | On-path | Heterogeneous | None | VSN | Improves caching in vehicular networks but faces redundancy in limited mobility environments. |
| Amadeo et al. (2022) [15] | PaC | Centrality-path | Heterogeneous | Closeness | NDN-IoT | Reduces retrieval delays but has high computational overhead for accurate closeness calculations. |
| Duan et al. (2022) [27] | DCAM | On-path | Homogeneous | None | ICN | Reduces loss recovery delays but relies on static content retrieval patterns, limiting adaptability. |
| Negara et al. (2023) [28] | C3CPS | Centrality-path | Heterogeneous | Betweenness | NDN | Effective multi-criteria decisions but computationally intensive for large networks. |
| Ali et al. (2021) [54] | NameCent | Off-path | Heterogeneous | Name Centrality | NDN | Mitigates data broadcast storms but lacks scalability in dynamic topologies. |
| Alduayji et al. (2023) [55] | PF-EdgeCache | On-path | Homogeneous | None | NDN | Efficient edge caching but increases computational complexity with freshness-based metrics. |
| Kumar & Tiwari (2023) [56] | DPPCOP | Partitioned Off-path | Heterogeneous | Distance-based Metric | CCN | Improves cache hit ratio but requires frequent partitioning adjustments in diverse applications. |
| He et al. (2024) [44] | EABC | Centrality-path | Heterogeneous | Betweenness | NDN-IoT | Balances energy efficiency and caching performance but struggles in latency-critical environments. |
| Amadeo et al. (2023) [45] | CCC | Centrality-path | Heterogeneous | Closeness | SDN | Reduces delays but introduces latency due to centralized SDN controller-based decisions. |
| Koide et al. (2024) [46] | ICANET | Centrality-path | Heterogeneous | Closeness | NDN | Reduces response delays but lacks real-time adaptability in dynamic networks. |
| Present Study | CacheCraft (Propose) | Centrality-path | Heterogeneous | PageRank | NDN | Achieves higher cache hit ratio and reduces latency with scalability in large, and dynamic networks. |

One recurring limitation in these studies is the assumption of homogeneous cache sizes across nodes, which neglects the varying traffic demands in real-world networks. Homogeneous allocation often results in inefficient resource utilization, frequent content replacement, and suboptimal cache hit ratios. The introduction of heterogeneous cache size dimensioning, guided by centrality metrics, has been identified as a critical step in overcoming these challenges.

CacheCraft, introduced in this study, builds upon this foundation by integrating PageRank centrality for centrality-path placement and heterogeneous cache size allocation. Unlike prior methods, CacheCraft dynamically adapts to changes in network topology, ensuring scalable and efficient caching performance in dynamic, large-scale networks.

### *2-5-2- Filling the Gaps with the CacheCraft Algorithm*

As highlighted in Table 1, most existing caching strategies have made strides in optimizing performance metrics such as cache hit ratios and latency but suffer from critical limitations:

1. ***Homogeneous Cache Size Allocation:***

Approaches such as DCAM [27], ERS [52], and PBCE [26] assume uniform cache sizes across nodes, leading to inefficient resource utilization and frequent content replacement in high-traffic areas. CacheCraft overcomes this by dynamically allocating heterogeneous cache sizes on the basis of PageRank centrality, optimizing resource usage.

2. ***Limited Adaptability in Dynamic Networks:***

Strategies such as NICE [48], and DCAM [27] perform well in static network topologies but lack adaptability to dynamic environments, such as IoT or mobile networks. For example, NICE relies on precomputed centrality metrics, which become outdated in scenarios with frequent topology changes. CacheCraft overcomes this limitation by dynamically updating node centrality values to adapt to network dynamics.

***3. High Computational Overhead:***

Centrality-based schemes, such as CMBA [29], and C3CPS [28] introduce significant computational complexity, particularly in large-scale networks. For example, C3CPS relies on multicriteria decision-making methods, which are computationally intensive. CacheCraft simplifies this process by focusing on PageRank centrality, which balances efficiency with scalability.

***4. Caching Redundancy:***

Redundancy in methods such as PBCE [26], where nodes with identical centrality values store the same content, leads to inefficient cache utilization. CacheCraft eliminates this by integrating connectivity and influence metrics within PageRank centrality for differentiated caching decisions.

***5. Trade-offs between Energy Efficiency and Performance:***

Approaches such as EABC [44] prioritize energy savings but compromise performance metrics such as latency and the cache hit ratio. CacheCraft balances energy efficiency and performance, making it suitable for IoT and large-scale network scenarios.

By addressing these limitations, CacheCraft demonstrates superiority through dynamic adaptability, improved resource utilization, and reduced computational overhead. Its effectiveness is validated in diverse topologies such as the GARR and Palapa Ring, which achieve higher cache hit ratios and lower latency than existing methods do.

## 3- Proposed Model: CacheCraft

The CacheCraft algorithm is designed to identify the storage needs of NDN routers and allocate capacity on the basis of their centrality values. By integrating seamlessly with the leave copy everywhere (LCE) placement strategy, CacheCraft requires minimal additional processing, most of which occurs during the initial network setup. Furthermore, CacheCraft dynamically adapts centrality values in response to changes in network topologies, ensuring that its caching strategy remains efficient and scalable. This adaptability significantly enhances the overall efficiency of Named Data Networking (NDN) systems. CacheCraft has emerged as an effective solution for NDN network operators, enabling more efficient network design and performance optimization to meet evolving traffic and content delivery demands.

Figure 2 illustrates the system model of CacheCraft. To allocate content store (CS) capacity to node i, the algorithm begins by calculating the node's connectivity via the L(j) symbol, which represents the number of links connected to the node. Additionally, the set of neighboring nodes directly linked to node i is identified via the RN symbol.

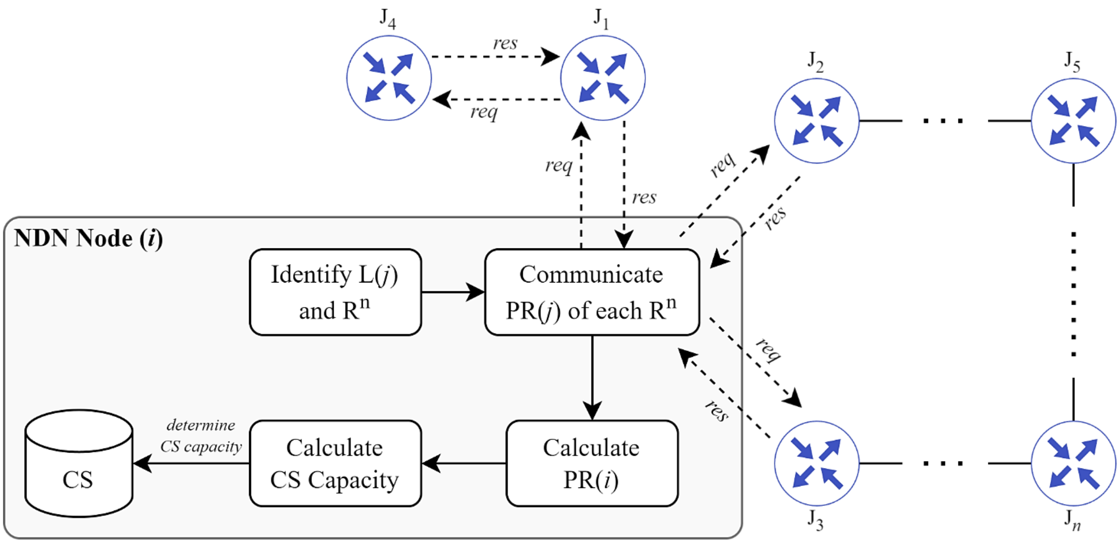



**Figure 2.** **CacheCraft System Model**

Next, each NDN node computes its PageRank (PR) value iteratively via Formula 5:

- In the first iteration (k = 1), all nodes are assigned an equal initial PageRank value, as no network connectivity is considered at this stage.
- For subsequent iterations (k > 1), each node communicates with its neighboring nodes to collect their respective PR(j) values. For example, as depicted in Figure 2, node i sends requests to its neighboring nodes (J1, J2, J3) to retrieve their PR values. Node-*i* then uses this information to update its own PageRank value. This iterative process is performed across all the NDN nodes until the centrality values converge.

The damping factor in the CacheCraft prevents the algorithm from becoming stuck in loop cycles and ensures convergence by balancing random node transitions and link-following behavior. The value of this factor ranges between 0 and 1. The closer to 1 the node transition behavior tends to continue following the existing links.

Once the PageRank (PR) values of all nodes have been determined, CacheCraft calculates the storage capacity (SC) for each cache node $R^c$ via Equation 6:

$$SC(i) = P \times c \times \frac{PR_{kmax}(i)}{PR_T} \quad s.t.\ i \in R^c \tag{6}$$

where $SC(i)$ is the optimal cache capacity for node i. $R^c$ is the set of cache nodes in the network. $P$ is the cache budget, which represents the total percentage of cache capacity distributed across the network. $c$ is the number of content prefix variations in the network. $PR_{kmax}(i)$ is the PageRank centrality of node i after kmax iterations. $PR_T$ is the total sum of all PageRank centralities, as defined by Equation 7**:**

$$PR_T = \sum_{i \in R^c} PR(i) \tag{7}$$

This formula ensures that the cache budget is allocated proportionally to each node's centrality, with highly connected or frequently accessed nodes receiving larger allocations.

Table 2 provides a comprehensive list of symbols, and their descriptions used in the CacheCraft algorithm, ensuring clarity and consistency in mathematical representations. These symbols are integral to defining the key parameters, intermediate values, and outputs of the proposed model.

**Table 2. Symbols are used to describe the algorithm**

| Symbols | Description |
|---|---|
| $n$ | Number of nodes in the network. |
| $c$ | Number of content prefix variations, representing the diversity of requested data. |
| $d$ | Damping factor, set to 0.85, balancing the probability of following links versus random jumps. |
| $kmax$ | Maximal iteration |
| $R^c$ | Set of cache nodes in the network. |
| $R^N$ | Set of neighbors directly connected to a given cache node |
| $PR_k$(i) | PageRank centrality of node-$i$ at the $\boldsymbol{k}$-th iteration |
| $PR_{k-1}$(j) | PageRank centrality of the $\boldsymbol{j}$-th neighbor node of $i$ at previous $k-1$ iteration |
| $PR(i)$ | Final PageRank centrality of node- $\boldsymbol{i}$ |
| $PR_T$ | Total sum of all PageRank centralities across the network. |
| $L(j)$ | Number of links connected to the $\boldsymbol{j}$-th neighbor node |
| $SC(i)$ | Optimal cache capacity of node- $\boldsymbol{i}$, representing the allocated Content Store (CS) size. |
| $P$ | Cache budget, the total percentage of cache capacity distributed across the network. |

The pseudocode for CacheCraft, shown in Algorithm 1, details the systematic process for calculating optimal cache sizes. This includes:

- Initialization: Assigning equal initial PageRank values to all nodes.
- PageRank Iterations: Update the PageRank values iteratively on the basis of neighboring nodes' contributions.
- Cache Size Calculation: Allocating cache sizes proportionally to the final PageRank centrality values.
- Output: Returning the optimal cache capacities for each router in the network.

This approach ensures efficient resource utilization while dynamically adapting to changes in network topology.

**Algorithm 1. Pseudocode of CacheCraft**

| **Input** | **Topology, Cache budget ($P$), Damping factor ($d$), and Maximal iteration ($kmax$)** |
|---|---|
| Output | Optimal cache capacity ($SC$) |
| 1 | for $i$ in $n$ do ← Calculate $PR(i)$ |
| 2 | for $k$ in $kmax$ do |
| 3 | if $k = 1$ do |
| 4 | $PR(i) = \frac{1}{n}$ |
| 5 | else |
| 6 | $PR_k(i) = \frac{(1-d)}{n} + d \times \sum_{j \in R^N} \frac{PR_{k-1}(j)}{L(j)}$ $\mathrm{PR_k(i)} = \frac{(1-\mathrm{d})}{\mathrm{n}} + \mathrm{d} \times \sum_{\mathrm{j} \in \mathrm{R^n}} \frac{\mathrm{PR_{k-1}(j)}}{\mathrm{L(j)}}$ |
| 7 | Calculate $\mathrm{PR_T}$ |
| 8 | for $i$ in $R^c$ do |
| 9 | Calculate $SC(i)$ |
| 10 | return Optimal content store capacity of each router |

### *3-1- Experimental Setup*

To evaluate the impact of the network topology and Network Centrality (NC) algorithms on the performance of Named Data Networking (NDN), we selected three real-world topologies: GARR, Palapa Ring, and Tiscali. These topologies represent diverse network structures and provide robust benchmarks for testing the proposed CacheCraft algorithm. This study assumes that the topology conditions are ideal (no node failures).

#### *3-1-1- Selected Network Topologies*

***1) GARR***

The GARR topology supports Italian national universities and research organizations. It consists of 61 nodes and 75 edges, including 27 cache nodes, 21 receiver nodes, and 13 source nodes [57]. This topology is characterized by high node density, with each node having a relatively small number of links. The maximum distance between a user and a server is 7 hops, with an average distance of 4 hops. This setup simulates a network with many dispersed connection points but limited connectivity per node.

***2) Tiscali***

Representing the Italian telecommunication network, Tiscali comprises 240 nodes and 404 edges, including 44 content source nodes, 36 receiver nodes, and 160 cache nodes [28]. This topology depicts a dense network where nodes are highly interconnected. The maximum hop distance between users and servers is 11 hops, with an average hop distance of 7 hops [58]. This topology models a network with high connectivity, presenting unique challenges in cache distribution.

***3) Palapa Ring***

The Palapa Ring models Indonesia's national fiber optic backbone network, which is designed to increase broadband accessibility across a vast region spanning 36,000 kilometers. It includes 83 nodes, comprising 10 content source nodes, 7 receiver nodes, and 66 cache nodes. This topology is notable for its distribution arrangement and uniform link distribution. The maximum hop distance between users and servers is 18 hops, with an average hop distance of 8 hops. Figure 3 illustrates the Palapa ring topology, and Table 3 provides detailed node specifications, including user, server, and cache node placements.

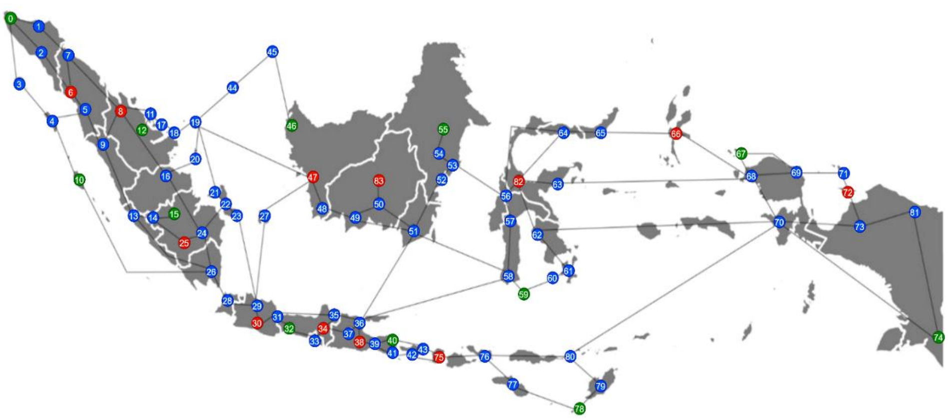


**Figure 3. Palapa Ring Topology**

**Table 3.** Specification Information for the Palapa Ring

| Device | Node |
|---|---|
| User | [0, 12, 15, 46, 10, 32, 40, 78, 74, 67, 59, 55] |
| Server | [6, 8, 25, 30, 34, 38, 75, 72, 66, 82, 83, 47] |
| Node NDN | [1, 2, 3, 4, 5, 7, 9, 11, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 31, 33, 35, 36, 37, 39, 41, 42, 43, 44, 45, 48, 49, 50, 51, 52, 53, 54, 56, 57, 58, 60, 61, 62, 63, 64, 65, 68, 69, 70, 71, 73, 76, 77, 79, 80, 81] |

### *3-1-2- Simulation Environment*

This study uses the Icarus simulator, which employs the leave copy everywhere (LCE) strategy for cache placement. This mechanism instructs nodes accessing content to retain a copy locally. For cache replacement, the least recently used (LRU) method is applied, ensuring efficient content eviction by replacing the least recently accessed content when the content store is full [59]. The experimental parameters are as follows:

- Cache capacity ratios: The cache budget ranges from 1% to 20% of the total content variety.
- Traffic distribution: The Poisson distribution is applied to represent the traffic request pattern in real- world cases.
- Content popularity distribution: A stationary Zipf's distribution with $\alpha = 0.8$ is used to represent the user's tendency toward content popularity in certain places. The greater the alpha value is, the more the content request pattern from users is concentrated on popular content.
- User behavior toward new content: In real- world cases, a user tends to explore new random content after finishing exploring the current content. Thus, $d$ in this study is 0.85. The damping factor $d = 0.85$ is selected on the basis of its proven accuracy in prior studies [60].
- Network Centralities: CacheCraft is compared against Uniform, Degree, Closeness, Eigenvector Centrality (EG), and Betweenness Centrality (BC) to benchmark its performance.
- Content Settings: The network contains 200,000 content items, with aggregate cache capacities set at 2,000, 10,000, 20,000, 30,000, and 40,000.
- Simulation Phases:
- Cache Warm-Up: This involves 100,000 content requests to initialize the system.
- Cache Measurement: Considers 400,000 content requests to evaluate performance.
- Metrics: Performance is assessed via the following:
- Cache Hit Ratio (CHR): The proportion of content requests served directly from the cache.
- Latency: The average delay in retrieving content.

Table 4 summarizes all the experimental parameters, including the topologies, centrality metrics, cache configurations, and workload distributions.

**Table 4.** Experimental Setup

| Parameter | Values |
|---|---|
| Topology | GARR, Palapa Ring, and Tiscali |
| Network Centralities | Uniform, Degree, EC, BC, Closeness, and CacheCraft (proposed method) |
| Cache Placement | LCE |
| Cache Replacement | LRU |
| Prefix Variations ($n_p$) | 200,000 |
| Cache Budget ($P$) | 1%, 5%, 10%, 15% and 20% |
| Warm-up Contents | 100,000 |
| Measured Contents | 400,000 |
| Content Popularity Distribution | Stationary Zipf |
| Request Traffic Distribution | Poisson |
| $\alpha$ | 0.8 |
| d | 0.85 |

### *3-1-3- Implementation Feasibility*

To validate CacheCraft, extensive simulations are conducted via real- world topologies, including the GARR, Palapa Ring, and TISCALI, ensuring that the results are grounded in practical scenarios. These topologies represent diverse and large-scale networks, offering a robust testbed for evaluating the algorithm under varying traffic loads and dynamic conditions. The simulations are designed to assess CacheCraft's performance in the following:

1. Latency reduction: By dynamically assigning cache sizes and strategically placing content.
2. Cache hit ratio: Through efficient utilization of network resources.
3. Computational efficiency: Leveraging PageRank centrality for scalable allocation.

Details of the simulation setup, parameter configurations, and evaluation metrics are provided in Section 4, along with a comparative analysis against state-of-the-art approaches.

## 4- Experimental Validation

This study evaluates the effectiveness of the proposed CacheCraft algorithm via three primary criteria: cache hit ratio (CHR), average latency, and time complexity. The performance of CacheCraft is compared with that of existing NC algorithms, including uniform, betweenness centrality (BC), degree, closeness, and eigenvector centrality (EG). Simulations are conducted under varying content store (CS) sizes, represented by different values of $P$.

### *4-1- Cache Hit Ratio (CHR)*

The Cache Hit Ratio (CHR) measures the efficiency of a caching system by calculating the proportion of successful cache lookups (hits) relative to the total number of cache lookups. It is computed via Equation 8 [9]:

$$CHR = \frac{n_{hit}}{n_{hit}+n_{miss}} \times 100\% \tag{8}$$

where $n_{hit}$ is the number of content requests successfully served by the cache node and where $n_{miss}$ is the number of content requests that failed to be served by the cache node. All types of content requested by users are considered when calculating CHR.

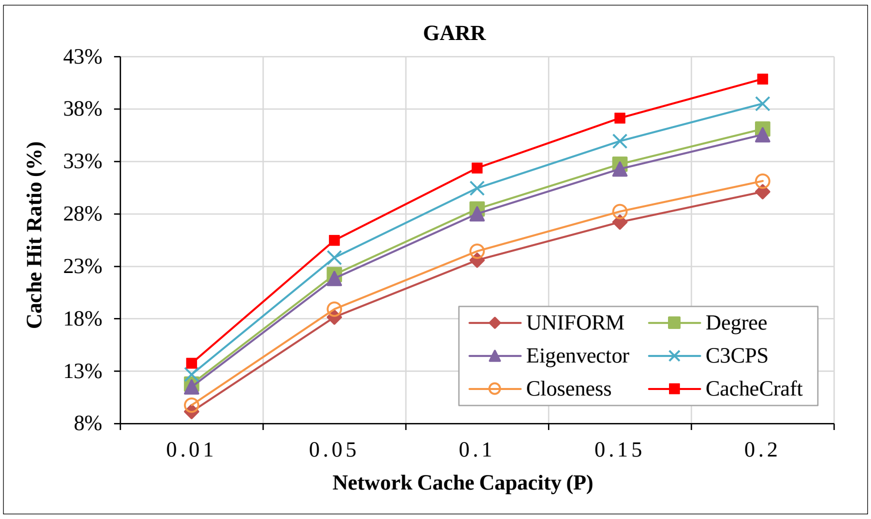
Figure 4 shows the simulation results for CHR across the three topologies: (a) GARR, (b) Palapa Ring, and (c) Tiscali. The results demonstrate that PageRank, as utilized in CacheCraft, outperforms all the comparison methods. A positive correlation is observed between increasing $P$, and the corresponding increase in CHR, as larger CS capacities enable NDN nodes to cache more content, reducing cache misses and improving CHR.

As depicted in Figure 4-a, the proposed CacheCraft algorithm consistently outperforms competing caching strategies in the GARR topology across all cache capacity levels, denoted by $P$. For instance, at $P$ = 0.01, CacheCraft achieves a Cache Hit Ratio (CHR) of 13%, surpassing the next best strategy, Betweenness Centrality (BC), by 2% and the uniform method by 4%. This performance gap widens as $P$ increases, with CacheCraft attaining a CHR of 40% at $P$ = 0.2, which is 10% greater than Uniform, 2.3% greater than BC, and 4.7% greater than Degree. Compared with C3CPS, CacheCraft’s reliance on PageRank centrality allows for more dynamic adaptability, ensuring that cache resources are allocated to high-traffic nodes in real-time, thereby avoiding the redundancy often observed in C3CPS’s static, multicriteria decision-making process.

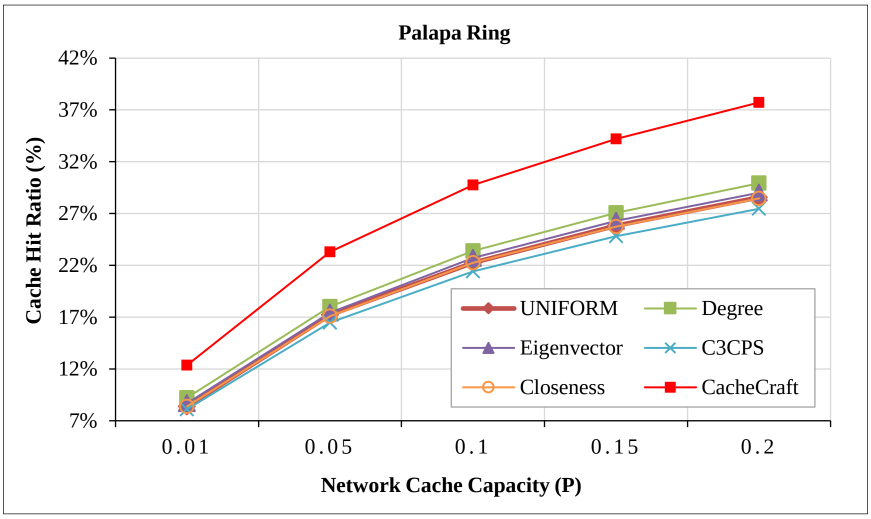
Figure 4-b highlights the performance of the Palapa ring topology. CacheCraft maintains its superior efficiency, achieving a CHR improvement of 9% over Uniform, 10% over BC, and 7% over Degree at $P$ = 0.2. Unlike C3CPS, which suffers from significant computational overhead due to its multicriteria decision-making methods, CacheCraft leverages the computational efficiency of the PageRank algorithm to scale effectively across complex topologies. This ensures not only higher CHR but also reduced latency, as CacheCraft allocates larger cache capacities to nodes with both strategic importance and higher traffic volumes.

Similarly, in the Tiscali topology, as shown in Figure 4-c, CacheCraft has a CHR that is 6.5% higher than uniform, 7.5% higher than BC, and 15% higher than Eigenvector Centrality (EG) at $P$ = 0.2. Compared with C3CPS, CacheCraft’s strategic placement of cache resources eliminates redundancy and prioritizes nodes on the basis of global influence, leading to consistently better performance across all metrics. C3CPS, constrained by its computationally intensive multicriteria approach, struggles to match CacheCraft’s adaptability and efficiency, particularly in dynamic network conditions.

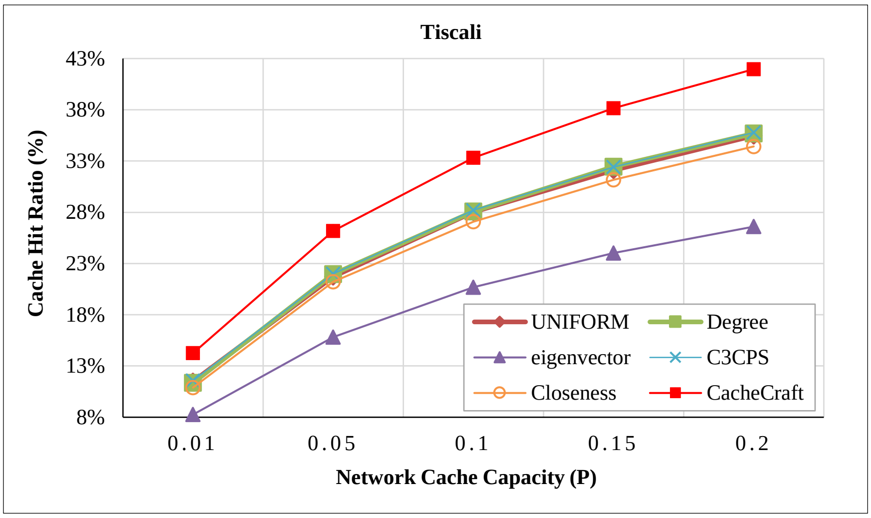
Overall, the proposed CacheCraft algorithm achieves superior CHR performance across all tested topologies and cache capacity levels by effectively balancing node significance, traffic volume, and network connectivity. Its ability to adapt dynamically, avoid redundancy, and efficiently allocate resources sets it apart from C3CPS and other traditional methods, makes it particularly suitable for large-scale and dynamic network environments.




(a)




(b)

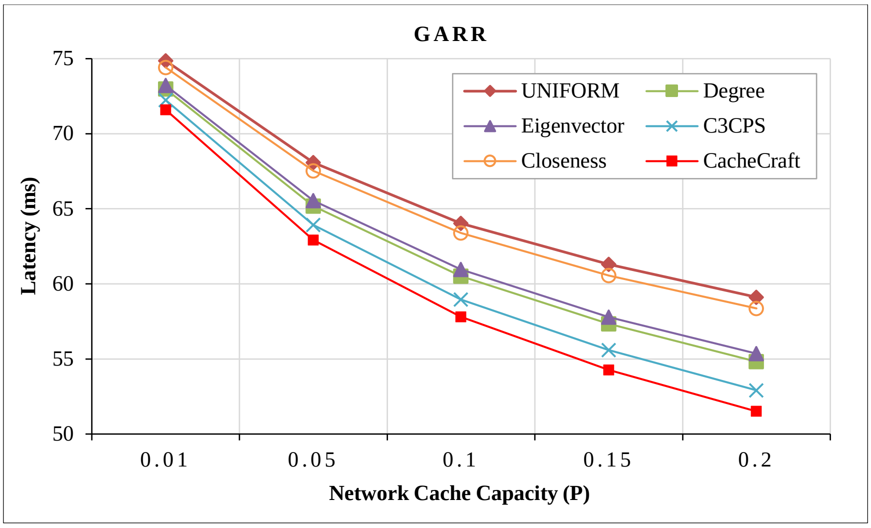



(c)

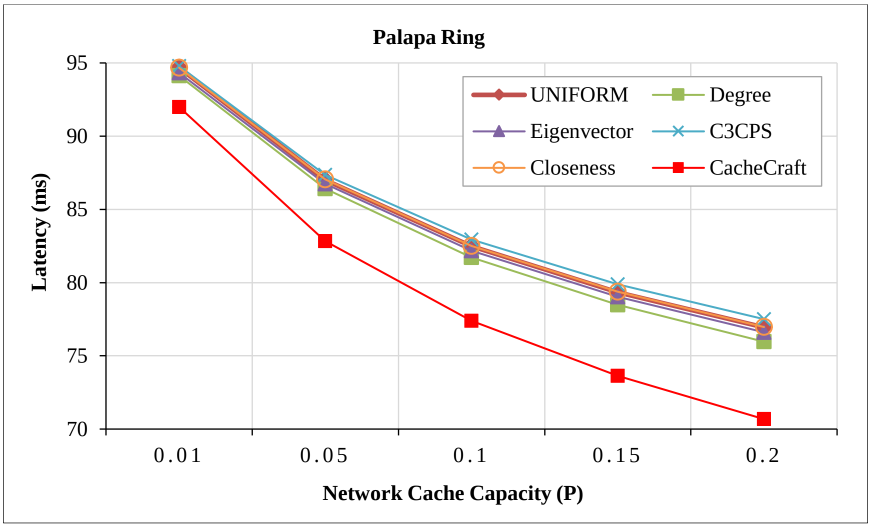
**Figure 4. Impact of Cache Capacity on the Cache Hit Ratio (CHR) across Topologies: (a) GARR, (b) Palapa Ring, (c) Tiscali**

### *4-2- Average Latency*

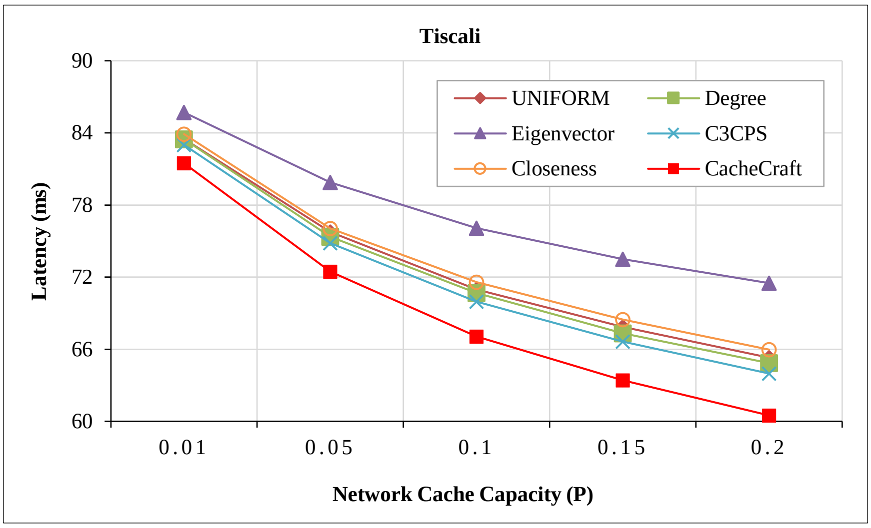
Latency represents the time taken between sending an interest packet and successfully receiving the requested data. A lower latency enhances the content distribution speed, whereas a higher latency slows the dissemination of materials. Latency is typically measured in milliseconds (ms) [10]. Figure 5 compares the performance of network centrality (NC) methods across three topologies (GARR, Palapa Ring, and Tiscali) in terms of latency. The CacheCraft results consistently show that CacheCraft achieves the lowest latency, followed by betweenness centrality (BC) in second place.




(a)




(b)




(c)

**Figure 5. Effect of Cache Capacity on the Latency (ms) across Topologies: (a) GARR, (b) Palapa Ring, (c) Tiscali**

Cache capacity impact on latency. As depicted in Figure 5, an increase in the cache budget ($P$) results in a corresponding decrease in average latency across all the topologies. This occurs because larger content store (CS) capacities allow NDN nodes to store more content, reducing the need for users to retrieve data from distant servers. Cache hits are more likely to occur on NDN nodes near the user, significantly lowering latency.

As shown in Figure 5-a, CacheCraft has the lowest latency across all the cache capacities ($P$) in the GARR topology. At $P$ = 0.2, CacheCraft achieves a latency of 51.5 ms, which is 2.6% lower than that of C3CPS (52.9 ms), 3.9% lower than that of degree (53.6 ms), 6.8% lower than that of closeness (55.3 ms), and 27.6% lower than that of uniformity (73.1 ms). The consistent reduction in latency is attributed to CacheCraft's use of PageRank centrality, which efficiently identifies and prioritizes high-traffic nodes, minimizing the distance between content and users. When $P$ increases, latency decreases for all the strategies because of the greater caching capacity, but CacheCraft consistently performs the best.

Similarly, Figure 5-b highlights the latency performance in the Palapa ring topology, where CacheCraft achieves a latency of 53.7 ms at $P$ = 0.2. This represents a reduction of 6.8% compared with C3CPS (57.6 ms), 7.1% compared with degree (57.9 ms), 6.3% compared with closeness (57.4 ms), and 7.4% compared with uniform (58.0 ms). CacheCraft's ability to allocate cache capacities dynamically on the basis of both node importance and traffic patterns ensures superior performance, even in complex topologies such as the Palapa Ring. The proposed scheme's adaptability to node centrality significantly reduces latency, as more frequently accessed content is cached closer to users.

Figure 5-c shows the latency results in the Tiscali topology, which represents a dense and highly interconnected network. At $P$ = 0.2, CacheCraft achieves a latency of 51.8 ms, which is 8.9% lower than that of C3CPS (57.0 ms), 6.2% lower than that of degree (55.2 ms), and 9.4% lower than that closeness (57.2 ms). The results underscore CacheCraft's ability to outperform all other strategies consistently by holistically considering both node connectivity and traffic demands when allocating cache resources. Unlike C3CPS, which often places cached content at nodes further from end-users due to static path assumptions, CacheCraft's dynamic and topology-aware approach effectively minimizes retrieval delays.

Overall, CacheCraft achieves the best latency performance in all topologies and cache capacities tested. By fully leveraging the PageRank algorithm, CacheCraft ensures that caching decisions account for both local and global network dynamics, leading to lower latency than traditional methods such as C3CPS, degree, and uniformity. This robust adaptability makes CacheCraft highly effective in reducing latency in both static and dynamic network environments.

### *4-3- Computational Time*

The computation time refers to the duration required by an algorithm to complete specific calculations or procedures. This metric directly correlates with the computational complexity of the algorithm, making it a critical factor in evaluating the scalability and efficiency of caching strategies. Figure 6 presents the computation time (in milliseconds) for various NC methods under different test conditions.

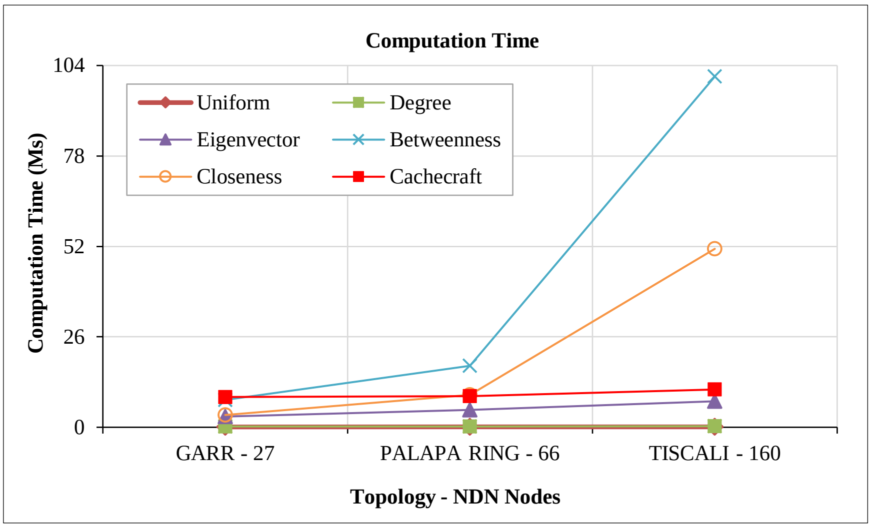



**Figure 6.** **Comparison of computation time (ms) across network centrality methods and topologies**

As shown in Figure 6, the computation times for betweenness centrality (BC) and closeness centrality increase exponentially with increasing network complexity. This is primarily due to the high computational demand of these algorithms, which involve calculating the shortest paths for all nodes in the network. Each shortest-path computation requires intricate calculations, making these methods less suitable for large-scale or complex networks. For instance:

- ***Betweenness Centrality***: The computational complexity is $O(nm + n^2 \log n)$. The computation time rapidly increases as the number of nodes ($n$) and the number of links ($m$) in the topology increase.
- ***Closeness Centrality***: While it evaluates the proximity of each node to all other nodes, this process involves calculating distances for every node in the network, further increasing the computational burden. However, the closeness computational complexity is $O(n(n + m))$.

In contrast, Uniform, Degree, Eigenvector, and CacheCraft exhibit algorithmic (linear or sublinear) growth in computation time, even in complex networks. These methods require fewer intensive calculations, allowing for greater scalability and efficiency:

- ***Uniform***: Assigns equal importance to all nodes without additional computations making it the most computationally efficient. This means the computational complexity is $O(n)$.
- ***Degree***: Node importance is computed by counting the number of links connected to each node whose computational complexity is $O(n+m)$. This straightforward calculation ensures low computational overhead.
- ***Eigenvector*** and ***CacheCraft***: Both rely on adjacency calculations to determine node centrality. While these computations are more intensive than uniform or degree computation are, they remain more efficient than BC and closeness computation are. CacheCraft, which leverages PageRank centrality, achieves scalable performance without compromising accuracy. The eigenvector complexity is $O(km)$ whereas the CacheCraft complexity follows the PageRank complexity which is $O(kn)$ where $k$ is the number of maximal iterations.

The results in Figure 6 highlight the limitations of betweenness (C3CPS) and closeness centrality in practical applications, particularly in intricate or large-scale networks. While theoretically robust, these methods demand extensive computation, rendering them impractical for real- world deployments requiring fast decision-making.

Conversely, CacheCraft demonstrates a balance between computational efficiency and effectiveness. By utilizing PageRank centrality, CacheCraft avoids the exponential growth in computation time observed in BC and closeness. Its reliance on iterative matrix calculations ensures scalability, making it well-suited for complex network scenarios. The performance of CacheCraft is comparable to that of the eigenvector in terms of computational efficiency but surpasses it in caching performance metrics, as discussed in earlier sections.

The findings validate CacheCraft as an efficient and scalable caching strategy that can be applied to real-world NDN deployments without introducing prohibitive computational delays.

## 5- Conclusion

This study introduced CacheCraft, a topology-aware caching framework designed to enhance named data networking (NDN) performance by optimizing content store (CS) capacity allocation. The framework uses the PageRank algorithm to determine the strategic importance of nodes and dynamically adjusts CS capacity allocations on the basis of traffic demands and network topology. Extensive evaluations of three topologies—GARR, Palapa Ring, and Tiscali—demonstrated that CacheCraft achieved an average improvement of 7.8% in the cache hit ratio (CHR) and a reduction in latency of 5.6 ms compared with state-of-the-art methods, including uniform, betweenness, degree, closeness, and eigenvector centrality-based approaches. The strengths of CacheCraft stem from its ability to allocate resources efficiently and adapt to real-time changes in network topology. By leveraging PageRank centrality, CacheCraft prioritizes nodes with greater connectivity and influence within the network, ensuring optimal performance with minimal computational overhead. This adaptability is particularly beneficial for large-scale and dynamic network environments, addressing critical issues such as high traffic load balancing and resource utilization efficiency. By addressing existing limitations in caching strategies and demonstrating robust empirical performance, CacheCraft paves the way for more adaptive and efficient NDN systems. Its contributions to improving network resource allocation, latency reduction, and cache performance mark a significant advancement in the field, with potential applications extending to various high-demand, dynamic network scenarios. This research also identifies pathways for further development. Future work to improve this study could include (i) integrating proactive caching strategies, such as machine learning models for content popularity prediction, to enhance CacheCraft's pre-emptive capabilities; (ii) adopting advanced cache replacement mechanisms informed by user behavior and content lifecycle data to further improve overall performance and scalability; and (iii) considering the impact of centrality toward topology conditions such as node failures.

## 6- Declarations

### *6-1- Author Contributions*

Conceptualization, R.M.N. and N.R.S.; methodology, R.M.N. and E.M.; software, R.M.N. and R.M.F.; validation, N.R.S., E.M., and M.S.B.; writing—original draft preparation, R.M.N.; writing—review and editing, R.M.F. and M.S.B.; visualization, R.M.N. All authors have read and agreed to the published version of the manuscript.

### *6-2- Data Availability Statement*

The data presented in this study are available on request from the corresponding author.

### *6-3- Funding*

This research is part of the ITB-NDN Research Group, funded by the Institut Teknologi Bandung, Indonesia.

### 6-4- Acknowledgements

We would like to acknowledge the support and facilities provided by the Institut Teknologi Bandung and Telkom University, Indonesia.


### 6-5- Institutional Review Board Statement

Not Applicable.

### 6-6- Informed Consent Statement

Not Applicable.

### 6-7- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

## 7- References


[1] Parrinello, E., Bazco-Nogueras, A., & Elia, P. (2024). Fundamental Limits of Topology-Aware Shared-Cache Networks. IEEE Transactions on Information Theory, 70(4), 2538–2565. doi:10.1109/TIT.2023.3321918.

[2] Zhao, Y., Zhang, S., & Yao, Z. (2023). A Hybrid Approach for Smart Alert Generation. International Conference on Electrical, Computer, Communications and Mechatronics Engineering, ICECCME 2023, 1–6. doi:10.1109/ICECCME57830.2023.10253233.

[3] Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., Claffy, K. C., Crowley, P., Papadopoulos, C., Wang, L., & Zhang, B. (2014). Named data networking. Computer Communication Review, 44(3), 66–73. doi:10.1145/2656877.2656887.

[4] Ur Rehman, M. A., Kim, D., Choi, K., Ullah, R., & Kim, B. S. (2019). A statistical performance analysis of named data ultra-dense networks. Applied Sciences (Switzerland), 9(18), 3714. doi:10.3390/app9183714.

[5] Lee, H., Lee, B., Yang, H., Kim, J., Kim, S., Shin, W., Shim, B., & Vincent Poor, H. (2023). Towards 6G Hyper-Connectivity: Vision, Challenges, and Key Enabling Technologies. Journal of Communications and Networks, 25(3), 344–354. doi:10.23919/JCN.2023.000006.

[6] Xia, X., Chen, F., He, Q., Cui, G., Lai, P., Abdelrazek, M., Grundy, J., & Jin, H. (2020). Graph-based data caching optimization for edge computing. Future Generation Computer Systems, 113, 228–239. doi:10.1016/j.future.2020.07.016.

[7] Pervej, M. F., Jin, R., Lin, S. C., & Dai, H. (2024). Efficient Content Delivery in User-Centric and Cache-Enabled Vehicular Edge Networks with Deadline-Constrained Heterogeneous Demands. IEEE Transactions on Vehicular Technology, 73(1), 1129–1145. doi:10.1109/TVT.2023.3300954.

[8] Alubady, R., Salman, M., & Mohamed, A. S. (2023). A review of modern caching strategies in named data network: Overview, classification, and research directions. Telecommunication Systems, 84(4), 581-626. doi:10.1007/s11235-023-01015-3.

[9] C.N., P., Vimala, H. S., & J., S. (2023). A systematic survey on content caching in ICN and ICN-IoT: Challenges, approaches and strategies. Computer Networks, 233, 109896. doi:10.1016/j.comnet.2023.109896.

[10] Yang, H., Pan, H., & Ma, L. (2023). A Review on Software Defined Content Delivery Network: A Novel Combination of CDN and SDN. IEEE Access, 11, 43822–43843. doi:10.1109/ACCESS.2023.3267737.

[11] Azamuddin, W. M. H., Aman, A. H. M., Sallehuddin, H., Abualsaud, K., & Mansor, N. (2023). The Emerging of Named Data Networking: Architecture, Application, and Technology. IEEE Access, 11, 23620–23633. doi:10.1109/ACCESS.2023.3243006.

[12] Rezazad, M., & Tay, Y. C. (2020). Decoupling NDN caches via CCndnS: Design, analysis, and application. Computer Communications, 151, 338–354. doi:10.1016/j.comcom.2019.12.053.

[13] Tanaka, D., & Kawarasaki, M. (2016). Congestion control in named data networking. 2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), 1–6. doi:10.1109/lanman.2016.7548848.

[14] Dinh, N. T., & Kim, Y. (2022). An Efficient Distributed Content Store-Based Caching Policy for Information-Centric Networking †. Sensors, 22(4), 1577. doi:10.3390/s22041577.

[15] Amadeo, M., Campolo, C., Ruggeri, G., & Molinaro, A. (2022). Popularity-Aware Closeness Based Caching in NDN Edge Networks. Sensors, 22(9), 3460. doi:10.3390/s22093460.

[16] Nour, B., Khelifi, H., Moungla, H., Hussain, R., & Guizani, N. (2020). A Distributed Cache Placement Scheme for Large-Scale Information-Centric Networking. IEEE Network, 34(6), 126–132. doi:10.1109/MNET.011.2000081.

[17] Yang, J., Liu, T., Yang, Y., Zhang, P., Wu, D., & Wang, R. (2022). Diversified Demands Integration-Based Proactive Caching Strategy towards Edge Networks. Wireless Communications and Mobile Computing, 2022, 1–14. doi:10.1155/2022/9378654.

[18] Ioannou, A., & Weber, S. (2014). Towards on-path caching alternatives in Information-Centric Networks. 39th Annual IEEE Conference on Local Computer Networks, 362–365. doi:10.1109/lcn.2014.6925792.

[19] Kumar, S., Tiwari, R., & Hong, W. C. (2021). Qos improvement using in-network caching based on clustering and popularity heuristics in CCN. Sensors, 21(21), 7204. doi:10.3390/s21217204.

[20] Gui, Y., & Chen, Y. (2021). A Cache Placement Strategy Based on Entropy Weighting Method and TOPSIS in Named Data Networking. IEEE Access, 9, 56240–56252. doi:10.1109/ACCESS.2021.3071427.

[21] You, H.-S., Kim, J.-H., Shin, W.-Y., & Kim, S.-W. (2021). Location-Aware Caching via Predicting Heterogeneous File Preferences in Mobile Networks. 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events (PerCom Workshops), 336–339. doi:10.1109/percomworkshops51409.2021.9431023.

[22] Freund, A. J., & Giabbanelli, P. J. (2022). An Experimental Study on the Scalability of Recent Node Centrality Metrics in Sparse Complex Networks. Frontiers in Big Data, 5. doi:10.3389/fdata.2022.797584.

[23] Chiranjeevi, M., Dhuli, V. S., Enduri, M. K., Hajarathaiah, K., & Cenkeramaddi, L. R. (2024). Quantifying Node Influence in Networks: Isolating-Betweenness Centrality for Improved Ranking. IEEE Access, 12, 93711–93722. doi:10.1109/access.2024.3424834.

[24] Wąs, T., & Skibski, O. (2023). Axiomatic characterization of PageRank. Artificial Intelligence, 318. doi:10.1016/j.artint.2023.103900.

[25] Cipolla, S., Durastante, F., & Tudisco, F. (2021). Nonlocal PageRank. ESAIM: Mathematical Modelling and Numerical Analysis, 55(1), 77–97. doi:10.1051/m2an/2020071.

[26] Lal, K. N., & Kumar, A. (2019). A popularity based content eviction scheme via betweenness-centrality caching approach for content-centric networking (CCN). Wireless Networks, 25(2), 585–596. doi:10.1007/s11276-017-1577-z.

[27] Duan, Y., Ni, H., & Zhu, X. (2022). A Dynamic Cache Allocation Mechanism (DCAM) for Reliable Multicast in Information-Centric Networking. Future Internet, 14(4), 1–15,. doi:10.3390/fi14040105.

[28] Negara, R. M., Syambas, N. R., & Mulyana, E. (2023). C3CPS: CRITIC-CoCoSo-based caching placement strategy using multi-criteria decision method for efficient content distribution in Named Data Networking. Journal of King Saud University - Computer and Information Sciences, 35(9), 101714. doi:10.1016/j.jksuci.2023.101714.

[29] Lal, K. N., & Kumar, A. (2018). A Centrality-measures based Caching Scheme for Content-centric Networking (CCN). Multimedia Tools and Applications, 77(14), 17625–17642. doi:10.1007/s11042-017-5183-y.

[30] Agryzkov, T., Tortosa, L., Vicent, J. F., & Wilson, R. (2019). A centrality measure for urban networks based on the eigenvector centrality concept. Environment and Planning B: Urban Analytics and City Science, 46(4), 668–689. doi:10.1177/2399808317724444.

[31] Li, Z., Tang, J., Zhao, C., & Gao, F. (2023). Improved centrality measure based on the adapted PageRank algorithm for urban transportation multiplex networks. Chaos, Solitons and Fractals, 167, 112998. doi:10.1016/j.chaos.2022.112998.

[32] Behera, R. K., Rath, S. K., Misra, S., Damaševičius, R., & Maskeliunas, R. (2019). Distributed centrality analysis of social network data using MapReduce. Algorithms, 12(8), 161. doi:10.3390/a12080161.

[33] Chai, W. K., He, D., Psaras, I., & Pavlou, G. (2013). Cache “less for more” in information-centric networks (extended version). Computer Communications, 36(7), 758–770. doi:10.1016/j.comcom.2013.01.007.

[34] Tortosa, L., Vicent, J. F., & Yeghikyan, G. (2021). An algorithm for ranking the nodes of multiplex networks with data based on the PageRank concept. Applied Mathematics and Computation, 392, 125676. doi:10.1016/j.amc.2020.125676.

[35] Wang, Y., Li, H., Zhang, L., Zhao, L., & Li, W. (2022). Identifying influential nodes in social networks: Centripetal centrality and seed exclusion approach. Chaos, Solitons and Fractals, 162, 112513. doi:10.1016/j.chaos.2022.112513.

[36] Ianni, M., Masciari, E., & Sperlí, G. (2021). A survey of Big Data dimensions vs Social Networks analysis. Journal of Intelligent Information Systems, 57(1), 73–100. doi:10.1007/s10844-020-00629-2.

[37] Chung, F. (2014). A brief survey of PageRank algorithms. IEEE Transactions on Network Science and Engineering, 1(1), 38–42. doi:10.1109/TNSE.2014.2380315.

[38] Hashemi, A., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2020). MGFS: A multi-label graph-based feature selection algorithm via PageRank centrality. Expert Systems with Applications, 142. doi:10.1016/j.eswa.2019.113024.

[39] Mendes, P. (2023). Named Service Networking as a Primer for the Metaverse. IEEE Communications Magazine, 61(9), 24–30. doi:10.1109/MCOM.005.2200713.

[40] Hou, J., Xia, H., Lu, H., & Nayak, A. (2021). A GNN-based Approach to Optimize Cache Hit Ratio in NDN Networks. 2021 IEEE Global Communications Conference (GLOBECOM), 1–6. doi:10.1109/globecom46510.2021.9685872.

[41] Amadeo, M. (2021). A Literature Review on Caching Transient Contents in Vehicular Named Data Networking. Telecom, 2(1), 75–92. doi:10.3390/telecom2010006.

[42] Hamidi, E. A. Z., Akbar, K. M., Negara, R. M., Payangan, I. P., Puspitaningsih, M. D., & As' Ari, A. Z. P. (2024, July). NDN Collaborative Caching Replacement and Placement Policy Performance Evaluation. In 2024 10th International Conference on Wireless and Telematics (ICWT), 1-5. doi:10.1109/ICWT62080.2024.10674708.

[43] Arif, M. S., Mukheimer, A., & Ejaz, A. (2024). Cannibalism and Harvesting in Tritrophic Chains: Insights from Mathematical and Artificial Neural Network Analysis. Emerging Science Journal, 8(4), 1262-1279. doi:10.28991/ESJ-2024-08-04-02.

[44] He, X., Liu, H., Li, W., Valera, A., & Seah, W. K. G. (2024). EABC: Energy-aware Centrality-based Caching for Named Data Networking in the IoT. 2024 IEEE 25th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 259–268. doi:10.1109/wowmom60985.2024.00049.

[45] Amadeo, M., Ruggeri, G., Campolo, C., & Molinaro, A. (2023). Content-Driven Closeness Centrality Based Caching in Softwarized Edge Networks. IEEE International Conference on Communications (ICC 2023), 3264–3269. doi:10.1109/icc45041.2023.10278928.

[46] Koide, M., Matsumoto, N., & Matsuzawa, T. (2024). Caching Method for Information-Centric Ad Hoc Networks Based on Content Popularity and Node Centrality. Electronics (Switzerland), 13(12), 2416. doi:10.3390/electronics13122416.

[47] Rossi, D., & Rossini, G. (2012). On sizing CCN content stores by exploiting topological information. 2012 Proceedings IEEE INFOCOM Workshops, 280–285. doi:10.1109/infcomw.2012.6193506.

[48] Khan, J. A., Westphal, C., Garcia-Luna-Aceves, J. J., & Ghamri-Doudane, Y. (2018). NICE. Proceedings of the 5th ACM Conference on Information-Centric Networking, 31–42. doi:10.1145/3267955.3267965.

[49] Luo, X., & An, Y. (2017). Neighbor cooperation based in-network caching for content-centric networking. KSII Transactions on Internet and Information Systems, 11(5), 2398–2415. doi:10.3837/tiis.2017.05.005.

[50] An, Y., & Luo, X. (2018). An In-Network Caching Scheme Based on Energy Efficiency for Content-Centric Networks. IEEE Access, 6, 20184–20194. doi:10.1109/ACCESS.2018.2823722.

[51] Zheng, Q., Kan, Y., Chen, J., Wang, S., & Tian, H. (2019). A Cache Replication Strategy Based on Betweenness and Edge Popularity in Named Data Networking. ICC 2019 - 2019 IEEE International Conference on Communications (ICC), 1–7. doi:10.1109/icc.2019.8761900.

[52] Delvadia, K., Dutta, N., & Ghinea, G. (2019). An Efficient Routing Strategy for Information Centric Networks. 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), 1–6. doi:10.1109/ants47819.2019.9118123.

[53] Meng, Y., Naeem, M. A., Ali, R., Zikria, Y. Bin, & Kim, S. W. (2019). DCS: Distributed caching strategy at the edge of vehicular sensor networks in information-centric networking. Sensors (Switzerland), 19(20), 4407. doi:10.3390/s19204407.

[54] Ali, I., & Lim, H. (2021). NameCent: Name Centrality-Based Data Broadcast Mitigation in Vehicular Named Data Networks. IEEE Access, 9, 162438–162447. doi:10.1109/ACCESS.2021.3133016.

[55] Alduayji, S., Belghith, A., Gazdar, A., & Al-Ahmadi, S. (2023). PF-EdgeCache: Popularity and freshness aware edge caching scheme for NDN/IoT networks. Pervasive and Mobile Computing, 91. doi:10.1016/j.pmcj.2023.101782.

[56] Kumar, S., & Tiwari, R. (2022). Dynamic Partitioning and Popularity based Caching for Optimized Performance in content-centric fog networks: DPPCOP. Pervasive and Mobile Computing, 88, 101740. doi:10.1016/j.pmcj.2022.101740.

[57] Chaudhary, P., Hubballi, N., & Kulkarni, S. G. (2022). NCache: Neighborhood Cooperative Caching in Named Data Networking. 5th International Conference on Hot Information-Centric Networking (HotICN), 36–41. doi:10.1109/hoticn57539.2022.10036203.

[58] Silva, A., Araujo, I., Linder, N., & Klautau, A. (2019). Name Popularity Algorithm: A Cache Replacement Strategy for NDN Networks. Journal of Communication and Information Systems, 34(2), 206–214. doi:10.14209/jcis.2019.22.

[59] Syambas, N. R., Situmorang, H., & Putra, M. A. P. (2019). Least Recently Frequently Used Replacement Policy in Named Data Network. IEEE 5th International Conference on Wireless and Telematics (ICWT), 1–4. doi:10.1109/icwt47785.2019.8978218.

[60] Langville, A. N., & Meyer, C. D. (2005). A survey of eigenvector methods for web information retrieval. SIAM Review, 47(1), 135–161. doi:10.1137/S0036144503424786.