International Journal of Advanced and Applied Sciences

Journal homepage : http://www.ijaas.in

A survey of classification cache replacement techniques in the content-centric networking domain

Sidra Batool [1], Muhammad Kaleem [1, *], Salman Rashid [2], Muhammad Azhar Mushtaq [1], Iqra Khan [1]

[1]Department of Information Technology, Faculty of Computing and Information Technology, University of Sargodha, Sargodha, Pakistan
[2]Department of Computer Science and Information Technology, University of Lahore, Lahore, Pakistan

ARTICLE INFO

Article history:
Received 3 December 2023
Received in revised form
11 April 2024
Accepted 19 April 2024

Keywords:
Content-centric networking
In-network caching
Cache replacement strategies
Content eviction policies
Efficiency optimization

ABSTRACT

Content-Centric Networking (CCN) is an innovative approach that emphasizes content. A key strategy in CCN for spreading data across the network is in-network caching. Effective caching methods, including content placement and removal tactics, enhance the use of network resources. Cache replacement, also known as content eviction policies, is essential for maximizing CCN's efficiency. When cache storage is full, some content must be removed to make room for new items due to limited storage space. Recently, several advanced replacement strategies have been developed to determine the most suitable content for eviction. This study categorizes the latest cache replacement strategies into various groups such as static, space scarcity, content update, centralized, energy-efficient, weighted, adaptive, and based on dynamic popularity. These categories are based on the approaches suggested in previous research. Additionally, this paper provides a critical analysis of existing methods and suggests future research directions. To the best of our knowledge, this is the most up-to-date and comprehensive review available on this topic.

1. Introduction

As the amount of digital content on the internet continues to grow exponentially, searching and retrieving specific content is becoming complex. Fulfilling the high demand for content from specific locations has become a challenge. The host-centric model, where the focus is on the location of the content, struggles to manage and distribute this massive amount of content. The need for content-centric networking (CCN) arises to handle the limitations of host-centric networks, such as latency, network congestion, and scalability (Wang and Ning, 2022).

The concept of CCN emerged as an alternative to the traditional host-centric networking model. The idea was first introduced in the mid-2000s in the context of rapidly growing content distribution and the rise of new-generation applications as researchers recognized the limitations of IP-based host-centric architectures. The goal was to shift the focus from devices to the content itself (Serhane et al., 2021). This involved using content names or identifiers to request and retrieve data rather than relying on hosts' specific IP addresses.

CCN is a novel approach that prioritizes content retrieval based on its unique identifier rather than the traditional host-centric approach. Potential research on CCNs is ongoing to fulfill the challenges. There are several key gaps that still need to be addressed. To gain its benefits and shape its future, there is a need for continuous exploration and innovation in the CCN modules such as content forwarding, content security, cache management, content naming resolution, and congestion control, as shown below in Fig. 1.

Cache management module plays an important role in ensuring the efficient use of limited storage resources. Cache placement and replacement are two fundamental techniques of cache management (Jaber and Kacimi, 2020). Cache placement focuses on where to store copies of content within the network, while cache replacement involves determining which content items should be removed from the cache when new content needs to be stored in limited cache space (Yu et al., 2020). The cache management module is a critical area of investigation.

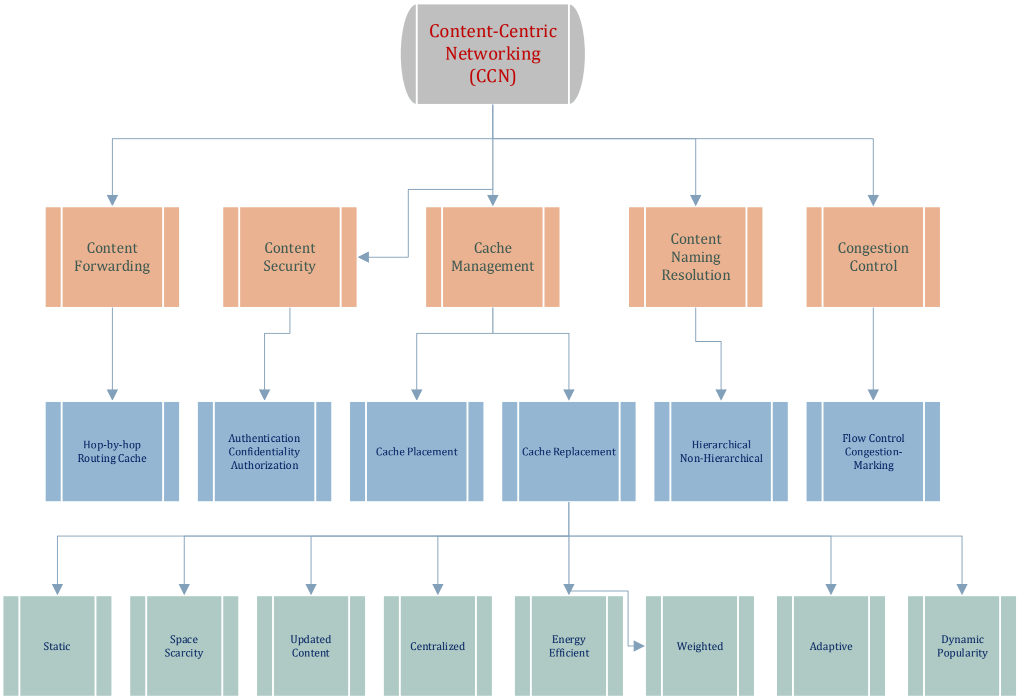

Fig. 1: Taxonomy of CCN

In CCN, the process of content forwarding includes sending Interest Packets from one router to another, searching for the requested content at each hop. When a user requests specific content, an Interest Packet is sent into the network. If a router's local cache already has the content (a cache hit), it responds by sending back a Data Packet containing the requested content. If the content is not in the cache (a cache miss), the router checks its Pending Interest Table (PIT), which records Interest Packets that have been previously sent upstream towards the content source. If there is no matching entry in the Content Store (CS) or PIT, the Interest Packet is forwarded based on the information in the Forwarding Information Base (FIB) and a new entry is created in the PIT. If there is a matching entry in the PIT, the request is added to the existing record, and the Interest Packet is discarded (Alahmadi, 2021). The Data Packet from the content source or an intermediate cache is then sent back along the reverse path. Whether an intermediate node stores this content in its cache or simply forwards it depends on the caching policy (Fig. 2).

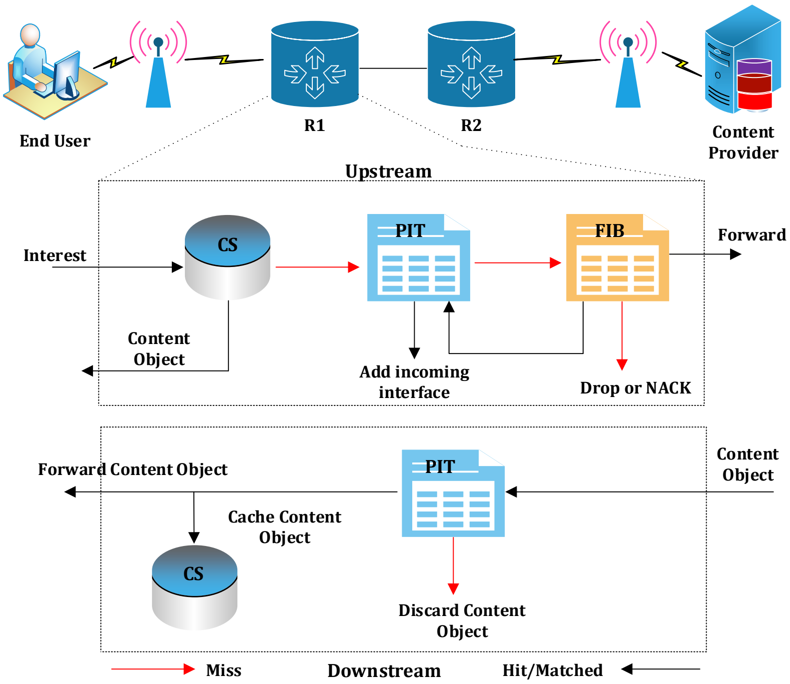

Fig. 2: Forwarding process at CCN node

Content security in CCN involves comprehensive measures to ensure the validity and safety of content in the network (Adhikari et al., 2020). CCN uses various mechanisms, including data authentication through cryptographic techniques such as hash functions, digital signatures, and certificates, to verify that the content's source is trustworthy. Data encryption is another security mechanism used in CCN to protect the privacy and security of information as it is exchanged across the network. Furthermore, data authorization in CCN involves setting specific permissions and policies to control access to content, ensuring that only authorized entities can access specific content.

Content naming resolution in CCN is a fundamental process that relies on unique content names for identifying and accessing data objects rather than specifying a specific server or IP address. It is the process of mapping content names to their corresponding data objects. In hierarchical content naming, the content namespace is divided into segments. Each segment represents a level of hierarchy, making it easier to manage and cache content at various levels (Shah et al., 2022). On the other hand, non-hierarchical content naming, also called flat naming, relies on unique identifiers typically generated using cryptographic hash functions. These are based on the contents' actual characteristics and eliminate the need for hierarchical structures. Congestion control in CCN is a content-driven mechanism to manage and reduce network congestion. For this purpose, a flow control mechanism is employed to manage the rate of data transmission, preventing congestion and maintaining optimal utilization of network resources. Additionally, the congestion marking process is used by CCN to mark/indicate the packets that encountered congestion (Nikmard et al., 2022). This marking can be done using various mechanisms, helping to maintain efficient data delivery. Overall, congestion control in CCN plays a significant role in managing congestion more effectively, even during periods of high network load. Cache placement in CCN is the process of strategically deciding where to store content to optimize content delivery and minimize data retrieval latencies (Jaber and Kacimi, 2020). The main objective is to maximize cache hit rates, thereby enhancing overall network performance. CCN employs various cache placement strategies to make decisions on where to store content based on the network's topology and content demand patterns. Cache replacement in CCN is the process of selecting the content to be removed from the cache when the cache reaches its capacity, and new content needs to be stored. Caches have limited capacity, and as new content is requested, some existing content must be replaced to accommodate the new entries. Numerous scholars have proposed cache replacement strategies to manage the caches within the network, which plays a significant role in influencing network performance (Victoria Priscilla and Charulatha, 2023). The replacement policies fall under several categories, including static, dynamic popularity, updated content, centralized, energy efficient, weighted, adaptive, and space scarcity.

2. Literature survey methodology

The systematic literature review undertaken aimed to categorize cache replacement techniques within CCN. The information derived from primary sources was organized and subjected to rigorous analysis. This process not only ensured a comprehensive understanding of the current state of cache replacement techniques in CCNs but also facilitated the categorization and classification of these approaches.

2.1. Research questions

The research questions serve as the guiding framework for the exploration and analysis of existing literature. The specific research queries devised for the current systematic literature review were as follows:

Q1: What is the impact of cache replacement techniques on overall network performance?
Q2: How can cache replacement techniques be categorized based on their underlying principles within content-centric networks?

2.2. Search resources

To explore cache replacement techniques in CCN, we pursued a comprehensive investigation using reputable search engines such as IEEE, Scopus, Springer, ACM, and Google Scholar (Fig. 3). Initially, we excluded root research materials unrelated to the main subject. Subsequently, we conducted a thorough analysis of selected research articles and conference proceedings, aligning our assessment with specific criteria. This study focused exclusively on academic works authored in English.

2.3. Selection criteria

Our literature survey's selection process included choosing search terms related to our research topic, focusing on recent publications, assessing the credibility of sources, and considering a variety of viewpoints to thoroughly understand the topic. We excluded certain research articles based on their publication date and the information in their abstracts. The final selection of studies, detailed in Table 1, primarily includes research published between 2014 and 2023.

3. Classes of cache replacement techniques

3.1. Static

Static refers to the cache replacement policies that follow fixed rules for making cache replacement decisions. Their decision on which content to remove from the cache is not influenced by the content's popularity or access patterns over time. Some of the most popular cache replacement static policies are Random Replacement (RR), Least Recently Used (LRU), Least Frequently Used (LFU), and First in First out (FIFO). The Random Replacement (RR) policy randomly chooses the content for replacement (Pfender et al., 2018). However, this technique is simple to implement but not so efficient because the most demanded content may be evicted earlier. First in First out (FIFO) is based on first-come, first-serve policy (Yu et al., 2020). The content that arrives first will be evicted first. So, it deletes the oldest content from a content store when needed. However, it doesn't deal with the number of requests or content popularity. The Least Frequently Used (LFU) policy (Chen et al., 2019) keeps a record of the request frequency of each content in the cache. Upon receiving a replacement request, the content with the lowest number of requests is replaced. But LFU doesn't reflect the dynamic popularity of content. So when a content loses its popularity, it may overstay in the cache due to its previous popularity. The Least Recently Used (LRU) replacement policy keeps a record of the usage of each content in the cache (Paschos et al., 2018). When a replacement request is received, it replaces the content that has not been used for a long. Although it is easy to implement before replacing content, it doesn't consider the number of requests for the content.

Table 1: Search results

Source	Early search	Exclusion based on time	Exclusion based on abstract	Final selection
Google Scholar	52	25	18	9
IEEE Explorer	81	38	32	11
Scopus	68	33	29	6
Springer	55	21	32	2
ACM	43	18	22	3

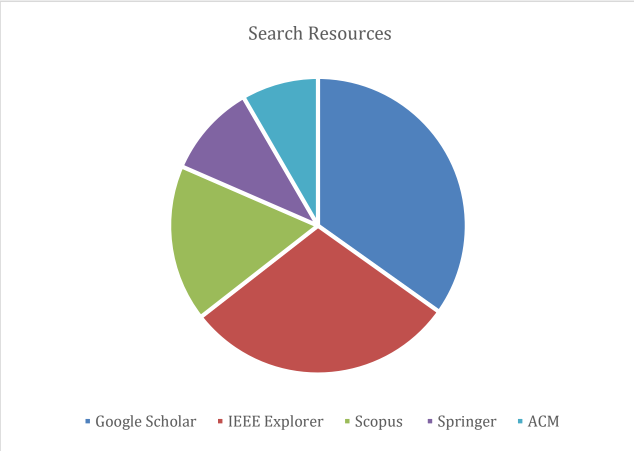

Fig. 3: Search outcomes

3.2. Space scarcity

Space scarcity refers to the limited storage capacity available in the network's cache. Once it becomes full, space scarcity becomes a concern. Therefore, CCN needs to make decisions for the efficient utilization of this limited space.

Ji et al. (2021) proposed a cache replacement strategy known as the content popularity and cache gain (PGR) method. This technique aims to make efficient use of limited cache capacity by reflecting the real-time popularity of content. The PGR policy determines content popularity based on how often and at what intervals consumers request the content. It places content in router nodes that are closer to consumers. To help estimate the cache gain, an additional field is added to both the interest packet and the data packet, which tracks the number of hops content travels during network transmission. Content popularity varies over time-based on the intervals between content requests, and cache gains help control the physical distance between consumers and the content. When new content arrives and the cache is full, less valuable content is removed. The value of content is calculated by taking the ratio of its popularity in the current cycle to the time interval between the current and previous requests and then adding the cache gain. Although this method considers cache gain, balancing between content popularity and cache gain can complicate eviction decisions; for example, a higher cache gain value might increase the perceived value of content with lower popularity. The gains and challenges of CCN content replacement policies are further discussed in Table 2.

Mishra et al. (2023) introduced a new content replacement policy in their research aimed at preserving important content within Information-Centric Networking-based IoT networks. This policy primarily addresses challenges such as heavy traffic mismanagement, scarcity of address space, and low efficiency. The policy considers three key characteristics of content: the total number of requests, the content's remaining lifetime, and the time elapsed since the last request (known as the last access time). These attributes are normalized to facilitate comparisons and evaluations. Based on the normalized values of these attributes, an eviction value is calculated for each piece of cached content. The content with the highest eviction value is selected for removal. This approach prioritizes removing less significant content (those that are rarely requested) to ensure that more valuable content is retained. However, the policy does not account for the dynamic popularity of the content, which can change over time.

3.3. Updated content

Updated content refers to the recent version of the content available in the network. In CCN, the content itself is the primary focus. Contents are kept updated to provide users with the most current information.

The least fresh first (LFF) algorithm as a replacement policy was proposed in Meddeb et al. (2019) research article. The objective of LFF policy is to address the challenge of keeping fresh data because cached contents can rapidly become outdated. A cached content is supposed to be valid if it has not been updated at the source since its last retrieval. By utilizing the ARMA time series model, the LFF algorithm estimates the remaining validity period (Tfresh) of cached content. This prediction is made using advanced forecasting tools, which are based not only on past events but also on some unexpected recent events. At the time of content eviction, the LFF algorithm employs a systematic approach to determine which content to remove from the cache. If all cached contents are still valid, the algorithm selects the one with the least remaining lifetime (Tfresh) for eviction. On the other hand, if multiple contents are already invalid, the LFF algorithm selects the one that has been invalid for the longest duration. This eviction strategy ensures that fresher data is retained in the cache, promoting the availability of up-to-date content for users. However, accurately estimating the freshness of cache blocks can be challenging in case of frequent cache updates. It may suffer from high cache pollution because infrequently accessed blocks may have high freshness values and occupy cache space, leading to lower cache hit rates.

Feng et al. (2022) introduced a popularity-based cache consistency management (PCCM) scheme to handle transient data from IoT devices in CCN. Traditional methods like time-to-live (TTL) and polling-every-time (PET) offer strong and weak consistency management across access routers (ARs) and content routers (CRs), respectively. However, these methods do not suit IoT's transient data well due to the limited cache sizes on routers along the data path. Therefore, tailored versions, T-TTL and T-PET, were also developed. Additionally, new fields such as version, generation time, and expected lifetime have been added to interest and data packets to aid cache consistency checks in CRs and ARs.

The effectiveness of these schemes was evaluated against the PCCM scheme using metrics such as hop counts, data staleness ratio, throughput, and signaling packets. The PCCM scheme, which adopts a hybrid approach, performed better on all these indicators. It treats frequently and infrequently updated data differently; for less popular data, data publishers maintain and communicate the lifetime of the data to ARs and CRs. For popular data, ARs periodically check with publishers for updates and notify the CRs along the path. However, the proposed model incurs a signaling overhead due to the introduction of new types of packets.

3.4. Centralized

Liu et al. (2021) introduced a novel intra-network cache replacement method that integrates content popularity and node betweenness for Information-Centric Networking (ICN) based IoT systems. This centralized cache management approach uses a central entity to oversee decisions on which content to evict from the cache, maintaining information about the overall network's cache status and the popularity of content.

Node betweenness is defined as the proportion of the total number of shortest paths through a network that passes through a particular node. The popularity of an edge router is calculated as the ratio of interest packets for a specific content received to the total number of interest packets received by the router. When calculating the popularity value, the next hop router normalizes these two metrics.

To manage cache status effectively while considering both content popularity and node betweenness, an intra-domain resource adaptation resolving server (RARS) has been established. This server keeps a synchronized popularity table with each router and updates betweenness values whenever there are changes in network topology. When intra-domain routers need to replace content in their cache, they compare the popularity values of the current and alternative contents using the popularity table to make a decision. However, this method requires ongoing synchronization between the RARS and each router, which can lead to additional overhead and increased network traffic.

3.5. Energy efficient

Energy efficiency is a significant concern in CCN. A primary goal is to reduce energy usage by making strategic decisions about which content to keep in caches (content retention) and which to remove (content eviction). An and Luo (2018) introduced a cache replacement scheme focused on improving energy efficiency within such networks. This method, known as the neighbor cooperation-based cache replacement (NCCR) technique, seeks to lower the overall energy costs associated with distributing content. Additionally, it aims to decrease the frequency of cache replacements, thus extending the duration for which content remains stored in network caches.

In this approach, when a caching node receives new content but lacks the space to store it, the least popular content is selected for migration. The neighboring node with the lowest accumulated cache load ratio is chosen as the destination for this content. If the neighboring node has sufficient cache space, the content is transferred there, and its details are recorded. If there is insufficient space, the least popular content in the neighbor's cache is replaced by the migrating content; if no space is available, the content is discarded by its original caching node.

Although this technique leverages the caching resources of neighboring nodes, coordinating the migration of content requires communication between them, which can lead to increased communication overhead.

Dinh and Kim (2022) introduced an energy reward-based caching (ERC) mechanism for IoT-based Information-Centric Networking (ICN), addressing the critical issue of energy efficiency due to resource constraints in IoT devices. Although existing caching strategies like Most Popular Content (MPC) and Compound Popular Content Caching Strategy (CPCCS) are effective in managing content, they are not always optimized for energy use, particularly in situations where IoT devices may have low or negative residual energy, preventing them from providing cached content to consumers.

The ERC mechanism operates on the principle that a content object is only cached if the residual energy of a node is equal to or exceeds a specified threshold. Additionally, a content object will replace an existing cached object if the energy benefit of doing so is positive; otherwise, the content may be cached on a different node along the forwarding path. The ERC model was compared with the MPC and CPCCS strategies in terms of energy efficiency, stretch ratio, and cache hit ratio. Through detailed analysis and experimentation, the ERC model demonstrated superior performance across these metrics.

However, the focus of this study is primarily on the energy efficiency of nodes, with less attention given to other critical performance aspects of IoT-based ICN. This could limit the broader applicability of the ERC model in scenarios where performance factors are also crucial.

3.6. Weighted

Weighted refers to the replacement approach that considers a set of factors contributing to cache eviction decisions. These factors are assigned different weights that reflect their importance in the cache replacement decision-making process.

Yu et al. (2020) proposed a cache replacement policy based on multi-factors named CRPM for named data networking NDN. The main purpose of introducing this scheme is to make effective use of cache resources by considering multiple factors contributing to cache eviction decisions. The cache value of content has been determined by analyzing multi-factors that affect the caching performance, which include content popularity, content acquisition energy consumption, content freshness, and the last access time interval. In order to ensure the effective use of cache resources, a new cache value function is created that makes the content with high value to be stored in the router for the longest amount of time possible. The content with the lowest cache value is removed from the caching space when the cache is full. As there are multiple factors involved in this technique, so calculating and maintaining these factors for each cache block can introduce additional overhead in terms of storage and computation.

The replacement strategy proposed by Alahmadi (2021) was the Enhanced Time and Frequency Cache Replacement Strategy (ETFCR). This policy has been introduced to improve the process of fetching and distributing content with efficient resource utilization. The proposed strategy is an enhanced Least Frequently Used (LFU). It consists of sorting the cache and calculating the weighted popularity of a chunk. It combines the request frequency and recent request time by taking into account time gaps between successive hits and adding with the current popularity. If there is too much time between hits, popularity will decrease, whereas it will increase if the requests are received quickly. If a replacement is required, the content with low popularity will be removed from the sorted cache. However, handling rapid changes in content popularity or access frequency might be challenging for the strategy to provide efficient cache management.

3.7. Adaptive

Adaptive refers to the approach that dynamically adjusts its content eviction decisions by switching among the fixed set of policies for different network situations.

The meta-policy replacement strategy, capable of learning appropriate replacement policies in different scenarios, has been proposed by Pires et al. (2022). The purpose of introducing this policy is to deal with the issue of replacement policies' varying efficiencies and to cope with the natural dynamism of context variations in the networks. Meta-policy strategy enhances the cache to work as an adaptive system, having the ability to learn the suitable policy among a fixed set of candidate policies for different network situations. Two procedures have been combined to propose a caching meta-policy strategy, i.e., a) Online Learning with Practical Feedback (OLPF) algorithm and b) Content and Context Management (CCM) module. OLPF is responsible for selecting policies to run at designated intervals while the CCM module manages the cache and measures cache efficiency. However, this technique requires a large processing time because it has continuous decision tasks at each iteration.

Singh et al. (2021) introduced an Adaptive Replacement Cache (ARC) policy designed for cache replacement in Named Data Networks. The goal of the ARC policy is to offer a scan-resistant approach that adapts effectively to changes in network traffic. The ARC policy is a self-tuning cache replacement strategy that dynamically adjusts to the changing conditions that impact cache efficiency. It does this by favoring either the Least Recently Used (LRU) or the Least Frequently Used (LFU) heuristic, depending on the current network traffic demands.

The ARC is adaptive and capable of modifying its behavior to suit various conditions. However, given that numerous factors can influence network traffic and there is no consistent pattern to predict the optimal policy for every situation, the ARC may not always be the best fit in all scenarios. This adaptability, while a strength, also introduces a level of uncertainty about its effectiveness in consistently meeting specific network demands.

3.8. Dynamic popularity

Dynamic popularity of content refers to how the interest and demand for specific content changes over time. This concept is critical in managing cache systems, where the goal is to avoid retaining unpopular content for too long.

Ji et al. (2021) introduced a replacement strategy called content popularity and cache gain (PGR), designed to efficiently use limited cache capacity and accurately reflect real-time content popularity. The PGR policy uses the frequency and timing of consumer requests to determine content popularity. It places content at router nodes closer to consumers and adds an extra field in the interest and data packets to record the number of hops content travels through the network. This data helps estimate the cache gain. The content's popularity may change over time and be influenced by request intervals, while cache gains manage the proximity of content to consumers. Content considered less valuable is removed from the cache to make room for new content when the cache is full. The value of the content is calculated by the ratio of its current popularity to the time interval between the current and previous requests, enhanced by the cache gain. However, balancing between content popularity and cache gain can complicate eviction decisions, such as when a higher cache gain increases the perceived value of less popular content.

Rashid et al. (2021) proposed another cache replacement policy for content-centric networks named Immature Used (IMU). This policy aims to address the issue of holding onto currently unpopular content and prematurely evicting potentially popular content. IMU calculates the content's period in the cache using its arrival time and frequency over a set duration. A maturity index is then determined by dividing the content's frequency by its period in the cache. A median of these maturity indices is used to assess content maturity; content with a maturity index above the median is considered mature, while below is considered immature. This concept helps guide eviction decisions based on the maturity levels of the content. The policy includes a time window check, where the time window size is 4, and the cache size is 6. When the time window expires, all content frequencies reset to 1.

Content is given time in the cache to establish its maturity level through a maturity classifier. When the cache is full and new content arrives, the least mature content is removed. Two algorithms are presented: one for determining content maturity and another outlining the IMU replacement policy. Performance was evaluated using the Icarus simulator, comparing results with other policies like FIFO, LFU, LRU, and LFRU, and placement strategies like Cache Less for More (CL4M), Leave Copy Everywhere (LCE), Leave Copy Down (LCD), and ProbCache. Various performance metrics, such as latency, cache hit ratio, path stretch, and link load, were measured and compared.

4. Case studies for content replacement in CCN

4.1. Case study 1

In a CCN scenario, a router embedded with a cache is designed to store frequently requested content. The cache management is facilitated by the implementation of the LRU algorithm. As users initiate requests for content, the router monitors the access history. When the cache reaches its capacity, the router employs the LRU algorithm to identify and evict the content that has remained unused for the longest duration, ensuring efficient utilization of the cache resources. LRU may not consistently represent the most optimal choice, as it might evict content that becomes popular again shortly after eviction.

4.2. Case study 2

In a Content-Centric Mobile Edge Computing (CC-MEC) ecosystem, mobile edge servers are responsible for maintaining caches that store frequently requested content from mobile devices. Currently, the cache management relies on a random replacement algorithm, wherein any randomly chosen item is to be evicted from the cache to make room for new content as incoming requests are processed. Random replacement algorithms demonstrate strong performance in environments characterized by dynamic and unpredictable content popularity. However, the absence of consideration for usage history could lead to suboptimal cache efficiency in specific scenarios.

These case studies highlight the significance of carefully choosing a cache replacement algorithm that aligns with the distinctive features of the CCN and anticipated access patterns. The optimal selection of a cache replacement algorithm is crucial for enhancing system performance and ensuring efficient content delivery in line with network dynamics and user behavior.

Table 2: Pros and cons of CCN content replacement policies

Category	Content replacement policies	Aim	Content eviction criteria	Pros	Cons	Simulation tool
Updated content	LFF (Meddeb et al., 2019)	Reduce content redundancy, ensure cache freshness	Content is evicted based on content freshness. It aims to better utilize cache space and improve network performance	Predict sensors' future events and maintain the data freshness	Computation overhead High replacement rate	ccnSIM
	PCCM (Feng et al., 2022)	Reducing IoT data that is outdated but cached by on-path routers	Based on the local residual caching time calculated for unpopular IoT data	Outdated IoT data can be removed timely by in-network caching	Signaling overheads	ndnSIM
Dynamic popularity	PGR (Ji et al., 2021)	Reflecting the current popularity of content in real-time	Content is evicted based on cache gain and current popularity	Recording the number of hops content is passing through and its current popularity	Balancing between content popularity and cache gain	ndnSIM
	IMU policy (Rashid et al., 2021)	To deal with the issue of overstaying unpopular content in cache	Least immature content gets removed from the cache	Maturity classifier to determine the maturity level of contents	Determine suitable size of cache and time window	Icarus
Adaptive	Meta-policy replacement strategy (Pires et al., 2022)	To deal with the issue of varying efficiencies of cache replacement policies	Depends on the current policy according to the network situation	Dynamically adjust to changing network conditions	Computational complexity due to continuous decision tasks at each iteration	ndnSIM
	ARC (Singh et al., 2021)	A scan-resistant approach for dynamically changing network	Depends on the current policy according to the network situation	Adaptive policy capable of adjusting its behavior	No explicit pattern to determine the best policy for specific context	ndnSIM
Weighted	CRPM (Yu et al., 2020)	To make effective resource utilization	Content that has the lowest cache value will be evicted	Contribution of multifactor such as content freshness, popularity, energy consumption, and access time	Computational and storage overhead	ndnSIM
	ETFCR (Alahmadi, 2021)	Efficient resource utilization	Content with low popularity value is removed	Request frequency and recent request time contributing to content popularity	Handling rapidly changing content popularity	ccnSIM
Energy efficient	NCCR (An and Luo, 2018)	To minimize energy consumption while making cache replacement decisions	Least popular content is evicted and chosen as migrating content	Utilizing the cache space of the neighboring nodes	Communication overhead due to coordination between neighboring nodes	ndnSIM
	ERC (Dinh and Kim, 2022)	To enhance caching in limited-resource IoT devices	Content with high energy consumption to cache is removed	Significant enhancement in terms of stretch ratio, energy efficiency, and cache hit ratio	Focuses only on energy efficiency aspect of IoT devices	COOJA
Centralized	PBRS (Liu et al., 2021)	To reduce data retrieval delays and improve the cache hit ratio	Content with low popularity value is removed	Intra-domain RARS maintains the cache status of network nodes	Network traffic overhead due to continuous synchronization between RARS and each node	ndnSIM
Space scarcity	PGR (Ji et al., 2021)	To efficiently utilize limited cache capacity	Content with less cache value is evicted	Content popularity in the current cycle, along with an extra field to record the number of hops content passes through the network (cache gain)	Balancing between cache gain and content popularity may lead to complex eviction decisions	ndnSIM
	Efficient content replacement policy (Mishra et al., 2023)	Developing cache replacement policy to run in low cache condition.	Content with least combined recency frequency (CRF) value is evicted	Integrating features of both LRU and LFU by Combined Recency Frequency value	The policy doesn't consider content validity and may delete fresh content	ndnSIM
Static	First in first out (FIFO) (Lee et al., 2013; Silva et al., 2022)	For scenarios where low computational overhead is desirable	Remove the content that has been stored the longest	Easy to implement, predictable behavior, and low overhead	No consideration of request frequency, limited context awareness, and potential for cache pollution.	ndnSIM
	Random replacement (RR) (Silva et al., 2022)	To introduce unpredictability and simplicity	Contents are randomly selected for eviction	Simple and unbiased	Most demanded content may be evicted earlier	ndnSIM
	Least frequently used (LFU) (Lee et al., 2013; Silva et al., 2022)	To consider the context of content access over time.	Content that is accessed infrequently is evicted	Prioritizing content items based on their access frequency	It doesn't reflect the dynamic popularity of the contents	ndnSIM
	Least recently used (LRU) (Lee et al., 2013; Silva et al., 2022)	To improve cache performance by giving priority to the most recently accessed content	Content that has been unused for a long is removed	Aligning the cache content with the current access patterns	It may not capture the long-term relevance of content	ndnSIM

5. Simulation tools

Recently, researchers have tried to develop an effective evaluation environment using simulators, emulators, and testbeds for CCN research (Hamdi et al., 2018). Simulators must be capable of handling all features and operations of ICN and providing in-depth analysis. Many simulators related to the ICN project exist, but some are largely used in research (CCNx, ccnSim, Icarus, ndnSIM). CCNx facilitates the operations of CCN, which is included in the CCN proposal. These operations focus on router design, end nodes, and security issues. Change the traditional OSI layer architecture of the internet in CCN. CCNx focuses on the interoperability between new layers in a real simulation testbed (Li et al., 2019); researchers pay less attention to transportation and caching operations. Therefore, alternative simulators have been developed to focus on caching operations and transportation. ndnSIM is another simulator that builds on top of NS-3 (Miglani and Kumar, 2023). So, it evaluates the behavior of transportation and basic caching operations in CCN. It is not appropriate to evaluate in-network caching. ccnSim is another packet-level simulator for evaluating caching behavior. It is built on Omnet++. It handles a large number of requests, different cache sizes, and a large content list. The trace-driven simulation does not support ccnSim. All the simulators under consideration incorporate support for caching, as it serves as a fundamental cornerstone in the majority of leading ICN proposals. Nevertheless, these simulators exhibit one or more inherent limitations that render them unsuitable when the primary objective is the assessment of caching performance.

Most existing ICN simulators have been specifically developed for the purpose of evaluating the performance of a singular architectural approach, but they lack the necessary extensibility to accommodate other architectures. The convergence of caching nodes to their steady state is characterized by a notably slow pace, contingent upon the prevailing traffic patterns. Consequently, simulations aimed at evaluating caching performance may necessitate a substantial volume of requests, ranging from several hundred thousand to tens of millions, to generate reliable results. Many ICN simulators have been primarily designed to address other facets, such as congestion control and protocol interoperability, which can generally be assessed through smaller-scale simulations. Consequently, these simulators have been implemented with a significantly lower level of abstraction than what is essential for caching, rendering them incapable of executing large-scale simulations within an appropriate time frame.

Network simulators play a crucial role in the field of computer networking. These assist researchers in evaluating systems before real-world implementations. Each simulator brings unique strengths, contributing to the advancement of networking technologies and protocols. Researchers often choose among the range of these tools based on their specific simulation requirements and focus areas. Commonly used simulators in the CCN literature are discussed, and their properties are presented in Table 3.

6. Simulation parameters used in the literature

In experimental setups, simulation parameters are employed to represent different facets of a process or system. These parameters can be modified to study their impacts on the simulated environment. Several cache-related parameters are typically used for experiments involving cache replacement policies. These include the size of the cache, the caching policy implemented, and the number of cache nodes. Additional parameters such as the duration of the simulation, the pattern of requests, the type of network connection, the routing strategy, and the characteristics of the source server are also taken into account. The specific simulation parameters referenced in related studies are detailed in Table 4.

7. Performance metrics used in the literature

Performance metrics are used to assess the quality, efficiency, and effectiveness of network operations and services. We can evaluate and compare the performance of multiple policies. Performance Metrics mentioned used in the literature are given below:

- Cache hit rate:

When a node gets a content request, and it is successfully satisfied from the cache, then it is called a cache hit. The cache hit rate is defined as the ratio of cache hits to the total number of attempted data retrievals (Yu et al., 2020).

$$Cache\ Hit\ Ratio = \frac{Cache_{hits}}{Cache_{hits}+Server_{hits}} \tag{1}$$

- Replacement frequency:

The contents of the cache are evicted due to limited cache space. So, the average number of cache replacement operations per second (how often the contents of the cache are replaced) in a node is called replacement frequency (Yu et al., 2020).

- Path stretch:

The number of hops traveled by a content request, from consumer to content provider/ publisher, is termed a path stretch. It quantifies the length or distance the request travels through the network's nodes before reaching its destination (Rashid et al., 2021).

$$Path\ Stretch = \frac{\sum_{i=1}^{n} Hop-Traveled}{\sum_{i=1}^{n} THop-Hop} \tag{2}$$

Table 3: Popular simulators comparison

Simulators	Replacement	Placement	CPU utilization	Language	Operating system
Icarus (Saino et al., 2014)	LRU, LFU, FIFO, RND, MIN, SLRU, CLIMB	LCE, FIX, LCD, BTW, ProbCach, HR, Edge, CLFM,	Low	Python	Linux
ccnSim (Tortelli et al., 2016)	LRU, LFU, FIFO, RND	LCE, FIX, LCD, BTW, ProbCach	Average	C++	Linux
ndnSim (Mastorakis et al., 2017)	LRU, LFU, FIFO, RND	LCE, FIX	High	C++ Python	Linux MacOS

Table 4: Simulation parameters

Simulation parameter	About	References
Cache size	Cache memory is used to temporarily store data. Its size is measured in bytes	(Ji et al., 2021)
Caching policy set	The set of rules to indicate how data should be stored and managed in cache	(Pires et al., 2022)
No. of cache nodes	Cache nodes are responsible for storing and serving cached data to clients	(Pires et al., 2022)
Content library size	Amount of digital content stored in a collection or library.	(Pires et al., 2022)
Simulation time	Elapsed time within a simulation	(Alahmadi, 2021)
Data freshness	Degree to which data is up-to-date or current at a given point in time	(Mishra et al., 2023)
File size	Size of a file on storage medium	(Alahmadi, 2021)
Warm-up requests	Sending preliminary requests to a server to prepare it for incoming traffic	(Rashid et al., 2021)
Model of popularity	Algorithm to characterize the popularity or demand for specific content items within the network	(Rashid et al., 2021)
Source server	A server that acts as the origin or source of data	(Ji et al., 2021)
Attenuation factor	The reduction in intensity, magnitude, or amplitude of a signal as it propagates through a medium	(Ji et al., 2021)
Link delay	Time it takes for data to travel from sender to receiver over a communication link	(Singh et al., 2021)
Interest rate	It represents how frequently consumers request content within the network	(Singh et al., 2021)
Network connection type	Connection type for establishing communication between devices (Wi-Fi, point-to-point)	(Mishra et al., 2023)
Routing strategy	A decision-making process to determine how data packets should be forwarded from source to destination	(Singh et al., 2021)
Payload size	Payload contains the content that the sender intends to transmit	(Singh et al., 2021)
Topology	It refers to how devices are connected to each other and how data is transmitted between them	(Rashid et al., 2021)
Publisher	Responsible for generating and making content available	(Mishra et al., 2023)
Subscribers	Users who access and consume content	(Mishra et al., 2023)
No. of simulation runs	Individual executions or instances of a simulation model	(Mishra et al., 2023)
Placement policy	Strategy used to determine where to store content to optimize content delivery and minimize data retrieval latencies	(Rashid et al., 2021)
Zipf distribution	Distribution of contents in a dataset in which a small number of items dominate while a large number of items have lower frequencies	(Yu et al., 2020)
No. of contents	Amount of information, material, or substance conveyed through diverse mediums	(Ji et al., 2021)
Content request pattern	Frequency and structure of requests made to access information.	(Pires et al., 2022)
Grid size	Size of topology that arranges nodes in a two-dimensional grid-like pattern	(Singh et al., 2021)
Operating System	Intermediary between computer hardware and user applications	(Putra et al., 2019)

• Latency:

The average time in which a system or service responds to a request OR the average time in which a content request is fulfilled and the content is received back by the consumer, is called the average response delay (Mishra et al., 2023).

$$Latency = Request\ Travel\ Delay + Content\ Travel\ Delay$$

• Content diversity:

Content diversity represents the amount of diverse content cached at a specific location. If the content redundancy is in the network cache, it will waste storage, and content diversity will be low. Content diversity refers to the variety of different content within a network, which impacts redundancy and overall network and cache performance (Naeem et al., 2020).

High content redundancy and low diversity can lead to increased latency and bandwidth usage, adversely affecting the network's efficiency. This includes degradation in the cache hit ratio and an increase in stretch, which is the lengthening of data paths within the network. Essentially, content diversity helps to minimize the presence of similar content across the network. The concept of content diversity is defined as follows:

$$\text{Diversity} = \frac{\sum(n(n-1))}{\mathrm{N}(\mathrm{N}-1)} \tag{3}$$

where: n refers to the amount of particular content type, and N represents the total contents of the network.

• Server hit reduction ratio:

The measure of reduction in the interest packets reaching the original content server is called the server hit reduction ratio. It indicates the percentage of requests being served by the cache, reducing the load on the origin server (Meddeb et al., 2019).

$$\text{Server Hit Reduction Ratio} = \frac{\text{Total requests to origin server} - \text{requests served by cache}}{\text{Total Requests to origin server}} \tag{4}$$

• Energy consumption:

Energy consumption refers to the amount of energy used by network devices such as routers or switches during their operation. This energy is expended in various activities, including caching, transmitting data, processing tasks, and retrieving content (Mishra et al., 2023).

• Validity:

It calculates the percentage of content successfully retrieved from cache nodes that aligns with consumers' expectations in terms of freshness and relevance. It reflects the network's ability to provide consumers with up-to-date data while minimizing outdated content (Meddeb et al., 2019).

$$\text{Validity (\%)} = \frac{\sum \text{Valid} * 100}{\sum \text{Valid} + \sum \text{invalid}} \quad (5)$$

• Link load:

The amount of traffic transmitted across the individual links in the network at a specific period of time. It accesses how heavily a specific network link is being utilized to transfer content between nodes, including both consumer requests and content retrieval (Rashid et al., 2021).

$$Link\ Load = \frac{(request_{size} \times request_{link\_count}) + (content_{size} \times content_{link\_count})}{Duration} \quad (6)$$

$$Duration = Content\ Retrieval\ Time - Content\ Request\ Time \quad (7)$$

• Energy saving rate:

The energy saving rate quantifies the efficiency of energy conservation within the network. It is calculated as the ratio of the energy saved through in-network caching, where content is stored and retrieved locally within the network, to the total energy consumption incurred when caching is not employed (An and Luo, 2018).

8. Discussion and future directions

The cache replacement policy is a critical component for caching in CCN. This policy has evolved from traditional host-centric caching to focus more on content-centric approaches. Unlike traditional methods that cache based on server or host identity, CCN caches are specifically designed to store and retrieve content based on its name. This approach leverages the hierarchical structure of content names, enabling efficient caching decisions that are informed by the names' structures and semantics.

CCN operates in a decentralized manner, allowing each router or node to independently manage its cache. This introduces specific considerations for cache replacement strategies, including the need to handle content freshness and manage different versions of the same content. CCN allows routers to make informed caching decisions by considering contextual information and the content's metadata and semantics. Crafting effective replacement rules is crucial for optimizing network performance and making efficient use of cache space, given the limited caching capacity of CCN nodes.

Replacement policies need to accommodate the network's unique characteristics, including user-specific access patterns, to tailor more personalized and efficient caching strategies. It is vital to implement adaptive policies that can dynamically respond to changes in network conditions, content popularity, and user behavior. Factors such as content size and network topology also play significant roles in the design of these policies.

Looking ahead, the future of cache replacement in CCN is set to be shaped by several emerging technologies. Machine learning could greatly enhance cache replacement strategies by analyzing historical access data to predict future content demands. As edge computing and Multi-Access Edge Computing (MEC) continue to grow, cache strategies will need to adjust to optimize content delivery at the edge. Additionally, dynamic network slicing, content-centric Quality of Service mechanisms, and the rollout of 5G networks are expected to influence cache replacement policies to cater to varying application needs. Integration with Software-Defined Networking (SDN), a focus on energy efficiency, efforts toward standardization, and the implementation of privacy-preserving measures are likely to play critical roles in the evolution of CCN cache replacement strategies. Furthermore, integrating intelligent caching algorithms could improve data retrieval and storage efficiency, enhancing content delivery and reducing latency in a network increasingly influenced by blockchain technologies and data dissemination demands. These advancements aim to optimize content delivery, enhance security, and align with the evolving landscape of modern networking technologies.

9. Conclusions

This survey paper delivers a comprehensive analysis and classification of contemporary cache replacement policies within CCN. CCN is a network architecture that is centered around content and data, prioritizing efficient content retrieval and distribution. The paper presents a taxonomy of CCN, which is organized into various modules, including content forwarding, content security, cache management, content naming resolution, and congestion control. The majority of replacement strategies discussed focus on factors such as content popularity, frequency, freshness, energy efficiency, effective cache space utilization, adaptability, and cost models. These cost models are influenced by a range of factors, such as network load, the geographical distance between publisher and consumer, and the reliability of the node.

The primary focus of this work is on the cache management module, where cache replacement policies are categorized into several types: static, space scarcity, updated content, centralized, energy-efficient, weighted, adaptive, and dynamic popularity. Additionally, the paper outlines the network simulators, simulation parameters, and performance metrics discussed in the existing literature. This survey aims to serve as a valuable resource for researchers, network engineers, and practitioners who are keen on gaining insights into the latest trends and techniques in cache replacement policies, assisting them in making informed decisions in the CCN domain.

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Adhikari S, Ray S, Obaidat MS, and Biswas GP (2020). Efficient and secure content dissemination architecture for content centric network using ECC-based public key infrastructure. Computer Communications, 157: 187-203. https://doi.org/10.1016/j.comcom.2020.04.024

Alahmadi S (2021). A new efficient cache replacement strategy for named data networking. International Journal of Computer Networks and Communications, 13(5): 19-35. https://doi.org/10.5121/ijcnc.2021.13502

An Y and Luo X (2018). An in-network caching scheme based on energy efficiency for content-centric networks. IEEE Access, 6: 20184-20194. https://doi.org/10.1109/ACCESS.2018.2823722

Chen L, Song L, Chakareski J, and Xu J (2019). Collaborative content placement among wireless edge caching stations with time-to-live cache. IEEE Transactions on Multimedia, 22(2): 432-444. https://doi.org/10.1109/TMM.2019.2929004

Dinh N and Kim Y (2022). An energy reward-based caching mechanism for information-centric Internet of Things. Sensors, 22(3): 743. https://doi.org/10.3390/s22030743 **PMid:35161490 PMCid:PMC8840271**

Feng B, Tian A, Yu S, Li J, Zhou H, and Zhang H (2022). Efficient cache consistency management for transient IoT data in content-centric networking. IEEE Internet of Things Journal, 9(15): 12931-12944. https://doi.org/10.1109/JIOT.2022.3163776

Hamdi MM, Habbal A, Zakaria NH, and Hassan S (2018). Evaluation of caching strategies in content-centric networking (CCN) for mobile and social networking environment. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 10(2-4): 1-6.

Jaber G and Kacimi R (2020). A collaborative caching strategy for content-centric enabled wireless sensor networks. Computer Communications, 159: 60-70. https://doi.org/10.1016/j.comcom.2020.05.018

Ji Y, Zhang X, Liu W, and Zhang G (2021). Replacement based content popularity and cache gain for 6G content-centric network. Physical Communication, 44: 101238. https://doi.org/10.1016/j.phycom.2020.101238

Lee J, Lim K, and Yoo C (2013). Cache replacement strategies for scalable video streaming in CCN. In the 19th Asia-Pacific Conference on Communications, IEEE, Denpasar, Indonesia: 184-189. https://doi.org/10.1109/APCC.2013.6765939

Li H, Zhou H, Quan W, Feng B, and Zhang H (2019). CCNHCaching: A high-speed caching throughput simulator for information-centric networks. Journal of Internet Technology, 20(3): 705-715.

Liu Y, Zhi T, Zhou H, and Xi H (2021). PBRS: A content popularity and betweenness based cache replacement scheme in ICN-IoT. Journal of Internet Technology, 22(7): 1495-1508. https://doi.org/10.53106/160792642021122207004

Mastorakis S, Afanasyev A, and Zhang L (2017). On the evolution of ndnSIM: An open-source simulator for NDN experimentation. ACM SIGCOMM Computer Communication Review, 47(3): 19-33. https://doi.org/10.1145/3138808.3138812

Meddeb M, Dhraief A, Belghith A, Monteil T, Drira K, and Mathkour H (2019). Least fresh first cache replacement policy for NDN-based IoT networks. Pervasive and Mobile Computing, 52: 60-70. https://doi.org/10.1016/j.pmcj.2018.12.002

Miglani A, and Kumar N (2022). Blockchain-based co-operative caching for secure content delivery in CCN-enabled V2G networks. IEEE Transactions on Vehicular Technology, 72(4): 5274–5289. https://doi.org/10.1109/TVT.2022.3227291

Mishra S, Jain VK, Gyoda K, and Jain S (2023). An efficient content replacement policy to retain essential content in information-centric networking based Internet of Things network. Ad Hoc Networks, 155: 103389. https://doi.org/10.1016/j.adhoc.2023.103389

Naeem MA, Rehman MAU, Ullah R, and Kim BS (2020). A comparative performance analysis of popularity-based caching strategies in named data networking. IEEE Access, 8: 50057-50077. https://doi.org/10.1109/ACCESS.2020.2980385

Nikmard B, Movahhedinia N, and Khayyambashi MR (2022). Congestion avoidance by dynamically cache placement method in named data networking. The Journal of Supercomputing, 78(4): 5779–5805. https://doi.org/10.1007/s11227-021-04080-0

Paschos GS, Iosifidis G, Tao M, Towsley D, and Caire G (2018). The role of caching in future communication systems and networks. IEEE Journal on Selected Areas in Communications, 36(6): 1111-1125. https://doi.org/10.1109/JSAC.2018.2844939

Pfender J, Valera A, and Seah WK (2018). Performance comparison of caching strategies for information-centric IoT. In the Proceedings of the 5th ACM Conference on Information-Centric Networking, ACM, Boston, USA: 43-53. https://doi.org/10.1145/3267955.3267966

Pires S, Ribeiro A, and Sampaio L (2022, April). A meta-policy approach for learning suitable caching replacement policies in information-centric networks. In the Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing, Association for Computing Machinery, Avila, Spain: 1950-1959. https://doi.org/10.1145/3477314.3507069

Putra MAP, Situmorang H, and Syambas NR (2019). Least recently frequently used replacement policy named data networking approach. In the International Conference on Electrical Engineering and Informatics, IEEE, Bandung, Indonesia: 423-427. https://doi.org/10.1109/ICEEI47359.2019.8988828

Rashid S, Razak SA, and Ghaleb FA (2021). IMU: A content replacement policy for CCN, based on immature content selection. Applied Sciences, 12(1): 344. https://doi.org/10.3390/app12010344

Saino L, Psaras I, and Pavlou G (2014). Icarus: A caching simulator for information centric networking (ICN). In the Proceedings of the 7th International Conference on Simulation Tools and Techniques and 7th International Conference on Simulation Tools and Techniques, ICST, Lisbon, Portugal, 7: 66-75. https://doi.org/10.4108/icst.simutools.2014.254630

Serhane O, Yahyaoui K, Nour B, and Moungla H (2020). A survey of ICN content naming and in-network caching in 5G and beyond networks. IEEE Internet of Things Journal, 8(6): 4081-4104. https://doi.org/10.1109/JIOT.2020.3022243

Shah MSM, Leau YB, Yan Z, and Anbar M (2022). Hierarchical naming scheme in named data networking for Internet of Things: A review and future security challenges. IEEE Access, 10: 19958-19970. https://doi.org/10.1109/ACCESS.2022.3151864

Silva ETD, Macedo JMHD, and Costa ALD (2022). NDN content store and caching policies: Performance evaluation. Computers, 11(3): 37. https://doi.org/10.3390/computers11030037

Singh P, Kumar R, Kannaujia S, and Sarma N (2021). Adaptive replacement cache policy in named data networking. In the International Conference on Intelligent Technologies, IEEE, Hubli, India: 1-5. https://doi.org/10.1109/CONIT51480.2021.9498489

Tortelli M, Rossi D, Boggia G, and Grieco LA (2016). ICN software tools: Survey and cross-comparison. Simulation Modelling Practice and Theory, 63: 23-46. https://doi.org/10.1016/j.simpat.2016.01.015

Victoria Priscilla C and Charulatha AR (2022). A deep dive comparison of cache replacement strategies: The quality of experience influencer. In the Computational Methods and Data Engineering: Proceedings of ICCMDE 2021, Springer Nature Singapore, Singapore, Singapore: 465-475. https://doi.org/10.1007/978-981-19-3015-7_34

Wang S and Ning Z (2022). Collaborative caching strategy in content-centric networking. In: Nicopolitidis P, Misra S, Yang LT, Zeigler B, and Ning Z (Eds.), Advances in computing, informatics, networking and cybersecurity: A book honoring professor Mohammad S. Obaidat's significant scientific contributions: 465-511. Springer International Publishing, Cham, Switzerland. https://doi.org/10.1007/978-3-030-87049-2_16

Yu M, Li R, and Chen Y (2020). A cache replacement policy based on multi-factors for named data networking. Computers, Materials and Continua, 65(1): 321-336. https://doi.org/10.32604/cmc.2020.010831