SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V13I4P104

Volume 13 Issue 4, 56-66, April 2026

Original Article

A²LFU: An Adaptive Ant Colony Optimization-BASED Hybrid Strategy for Efficient Caching in Content-Centric Networks

A R Charulatha[1*], Dr. C. Victoria Priscilla[2]

[1*]*Department of Computer Science, Stella Maris College (Autonomous), Chennai-86, Tamil nadu, India.*
[2]*Research Department of Computer Science, Shrimathi Devkunvar Nanalal Bhatt Vaishnav College for Women (Autonomous), Chrompet, Chennai-44, Tamil nadu, India.*

[1*]*Corresponding Author : charulatha@stellamariscollege.edu.in*

Received: 08 January 2026 Revised: 16 February 2026 Accepted: 17 March 2026 Published: 30 April 2026

Abstract - *The advancement of content-focused communication applications has made the Content-Centric Networking (CCN) appear, where network caching can play a key role to improve the efficiency and reduce redundancy in retrieving data. Classic approaches, such as LRU (Least Recently Used), LFU (Least Frequently Used), and ProbCache at the level of caching, have already been used. Of these, LFU is well known for its effectiveness, but it fails to be adaptive towards changes in content popularity as well as network topology under dynamic and real-time scenarios. In order to mitigate these issues, in this paper, a new caching method named "Fully Tuned Adaptive Ant Colony Optimization-based LFU Replacement" (A²LFU Hybrid) is proposed. The strategy proposed considers that both LFU and Ant Colony Optimization (ACO) are used jointly in order to make the policy adaptation more flexible for real-time network conditions, such as variation of content popularity. Performance analyses with traces from real traffic loads show that A²LFU Hybrid can achieve up to % higher average hit ratio and overall system performance than existing conventional methods, such as LFU, LFUDA, LRFU, etc., and recently proposed adaptive solutions. It also outperforms the other recent algorithms, such as Adaptive Ant Colony Optimization (ACO) and Adaptive Cuckoo Search.*

Keywords *- Content-Centric Networking (CCN), Ant Colony Optimization (ACO), Least Frequently Used, Adaptive Caching, Hybrid Strategy, Network Performance.*

1. Introduction

With the dramatic proliferation of web-based services, accompanied by a tendency for data traffic to double every 18 months, traditional network architectures are being stretched to their limits. Content-Centric Networking (CCN) is proposed as one promising approach to tackle these challenges by changing the traditional host-centric communication to content-centric communication. In CCN, the goal is to maximize content retrieval and delivery, and to achieve faster, more efficient, and scalable data retrieval. Caching, as an essential content store to maintain popular content close to users, is among the fundamental parts of CCN, providing enhancements in the network performance and reduction of latencies. Caching policies are decisive on the efficacy of content delivery since they influence the hit ratio, response time, and system performance. Although LFU is found to be more effective in some specific settings, it has one major drawback that restricts its use: it does not adapt over time according to the dynamics of the changes in popularity and network topology, which may differ significantly between a real and a synthetic setting. This restriction makes LFU much less efficient in the case of content popularity, which changes with time, such as Content-Centric Networks (CCNs).

Content-Centric Networking (CCN) is one of the network architectures that has been proposed in answering this demand, by focusing on content retrieval from large-scale networks with minimum delay. Compared to traditional host-based communication, CCN treats data differently by separating the content from its location, thus providing a scalable and flexible means for delivering content. Because of this characteristic, caching popular content near users in C-ICN can effectively improve the level of content retrievability and relieve the pressure on the network. This is particularly important in contemporary networks, such as 5G, IoT, and vehicular networks, where the content access patterns are volatile and nondeterministic. Mao et al. introduced the Hierarchical Content Caching and Asynchronous Updating (HCCAU) scheme for NT NACAVs, in their work on hierarchical content caching and asynchronous updating schemes for non-terrestrial network assisted connected automated vehicles [2]. [1] proposes a new cache mechanism in CCNs for VANETs and specifically for NTNs. This study has made it clear that caching is essential in the network with rapid mobility and directed topology, such as CAVs. The paper proposes a tree-based hierarchy of cache having an asynchronous update mechanism that is highly essential to offer data availability and latency in these highly demanded spheres. The hierarchical formulation allows us to make an optimal caching decision by considering the source locations of vehicles and their future requests for the contents, which is extremely desirable in the real-time adaptive caching of CCNs. The work by Mao et al. will underlie the future study of dynamic caching algorithms in mobile and changing networks, which we will address in this paper using adaptive AI algorithms like Ant Colony Optimization (ACO).

Khan et al. [2] suggest a cache-conscious and resource-saving content caching policy within the CCN architecture to minimize the data retrieval time with consideration of the network resource sharing property. They are concerned with latency and resource restrictions in a large-scale network with variable content demand. Their plan uses dynamic algorithms that dynamically handle the content placement based on the network load and the user request in order to enhance the cache and content delivery efficiency. This work is important for learning about the demand of policies that adapt to dynamic network conditions on-the-fly, which is in the philosophy of improving classical caching algorithms (such as LFU). The efficient techniques introduced in this paper will be useful to improve the assignment of resources for a caching solution, which justifies the choice of ACO as a way to adapt LFU-based caching for different scenarios.

Jungjit et al. [3] investigate the use of Ant Colony Optimization (ACO) in feature selection for multi-label correlation problems. Their contribution generalizes the application of ACO from classic network optimization problems and illustrates that it is applicable for finding good sets of features in feature-rich environments. The methods proposed in this paper, deploying ACO for optimizing the decision-making and resource allocation, are an ideal fit for cache placement. With ACO, this study can coordinate cache decisions by access pattern during caching and promote effective content retrieval in CCNs. This work serves as a guideline for the generic possibility of implementing ACO in other optimization realms, further encouraging the integration of ACO with LFU to form a more adaptive and intelligent caching policy. The address adaptive caching strategies in data networks, specifically employing bio-inspired optimization techniques, such as ACO. Their work is responsive to the increasing requirement of adaptation/elasticity of caching in today's networks, as it has to deal with dynamically varying content demand and network dynamics. They suggest an ACO-based bio-inspired algorithm in order to decide whether to cache or not with respect to the traffic patterns and system limitations. This is consistent with the motivation of this work to increase LFU using adaptive methods such as ACO. The contributions from this work show that bio-inspired algorithms can enhance caching efficiency, shorten retrieving time, and further utilize network resources for Content-Centric Networks (CCNs).

Zhao and Cao [4] focus on the use of extensive-form games for content caching in edge networks. They illustrate caching decisions as a strategic game between content providers and consumers, concentrating on the realization of efficient content delivery while considering the decentralized feature in edge networks. They emphasize the value of adding different objectives, such as lowering delay and increasing content availability, into solutions to cache content. While game-theoretical methods are effective for enhancing interactions in distributed networks, they are usually too computationally expensive. The work of Zhao and Cao emphasizes the importance of developing more effective techniques, such as ACO, for coping with large-scale dynamic workloads without invoking intensive computing overhead. Their work adds to the knowledge on network-level optimization and supports the use of hybrid approaches, able to capitalize on the advantages due to the adaptive feature of ACO and simplicity of traditional methods (i.e., LFU).

Awadallah et al. [5] have conducted extensive reviews of multi-objective Ant Colony Optimization (ACO) in different optimization problems, including content caching. Their work highlights how ACO can address difficult problems of high dimensionality by enabling parameters to be adapted on the fly based on real-time changes. The presented study shows that ACO is capable of providing desired flexibility and scalability in content caching for large-scale dynamic networks, which are those with great changes in user requests and network states. The research is necessary in order to design adaptive caching algorithms to suit different requirements in decision-making, such as cache hit ratio, latency, and resource consumption. These observations also inspired us to test AdaptiveCO and its performance in the proposed A2LFU Hybrid model, which also supports the design choice, since this fact demonstrates that the complex objective optimization capabilities of ACO can be adapted with CCNs.

Yin, Tu, Chen [6] introduce a novel tree-based method to aggregate data in a wireless sensor network by means of ACO and the cuckoo search algorithm. They optimize the paths of sensor networks' data collection by integrating ACO and cuckoo search to reduce the network energy consumption and enhance efficient communication. Their technique is flexible in resource-limited and data-flow dynamic environments, as evidenced by the deployment of ACO. They concentrate on sensor networks, although the same principles can be applied to help optimize resource allocation and enhance network efficiency for content caching in CCNs. It is also demonstrated in this work how it is possible to integrate ACO with other optimization techniques to produce effective caching strategies in general, which justifies the dual aspect of the hybrid strategy towards A2LFU Hybrid. Chen [7] explores how Ant Colony Optimization (ACO) information-centric networking delivery strategy is used to examine flow and scheduling to maximize content distribution in CCNs. In the Inheritance Content Chen presented a penetration demonstration to ACO in the content delivery policies, particularly the need for the content to be delivered based on the demands. The paper describes the contribution and novelty of ACO in terms of flexibility, scalability, and efficiency, which provides strong consideration for the enhancement of traditional caching algorithms such as LFU. This shows that introducing ACO improves a content-centric system in which the dynamic optimization of traffic delivery and cache-based decisions are supported, further motivating hybridizing LFU with ACO as in the proposed A²LFU Hybrid.

Faced with these problems, an increasing demand is being placed on caching schemes that are flexible enough to adapt dynamically to the varying state of both the network and the requested content. Recent developments have investigated the use of optimization algorithms, e.g., Ant Colony Optimization (ACO), to improve the flexibility and performance of caching mechanisms. ACO is a bio-inspired optimization algorithm that has demonstrated very good results in solving complex network design problems such as routing and load balancing, and recently also caching. ACO works on the principles of ants in nature, finding their way to the source cooperatively so that they can find the best path (least distance). By the introduction of ACO techniques, the cache strategy will become adaptive and optimize both the content storing and retrieving process dynamically according to network conditions as well as users' access patterns. This paper introduces a new hybrid caching strategy referred to as "fully tuned adaptive ant colony optimization-based LFU" Strategy (A²LFU Hybrid), which tries to combine the best of LFU and ACO together. The major issue tackled in this approach is how to mitigate the intrinsic weakness of LFU by adapting to changes in content popularity and network topology, thus enhancing its accuracy in practical situations. A²LFU Hybrid adopts ACO to adaptively refine the frequency-based caching decisions, ensuring popular content is properly stored and redundancies are minimized.

The proposed method is then validated through intensive simulations using traffic traces from reality. The experimental results show that A²LFU Hybrid outperforms conventional caching techniques, such as LFU and other adaptive strategies (i.e., Adaptive ACO-Adaptive Cuckoo Search). The hit ratio and the whole system performance are improved by the proposed approach, which improves content retrieval with high efficiency and reduces network latency in CCN. The rest of the paper is organized as follows: Section 2 introduces related work on content caching, which includes LFU and optimization-based solutions. The detailed design of the A²LFU Hybrid approach is described in Section 3. The simulation set-up and performance measures used in the evaluation are also described in Section 4. The results are reported (Section 5), and the performance of A2LFU Hybrid is compared with other caching methodologies. Lastly, Section 6 will provide a conclusion of the paper and discuss potential future research directions of adaptive caching within CCNs.

2. Literature Survey

In this part, gives a literature review of the recent developments on content caching schemes in CCNs. It reflects on the classical and AI-based methods, highlights key issues and solutions to optimize content delivery, resource management, and adaptive real-time. The review also points out the research gaps and potential directions for further improvement in terms of caching strategies. Asmat et al. [8] take into account the energy-saving methods of caching in the IoT Content-Centric Networking (CCN). They propose centralized caching, whereby content delivery might be optimal in terms of energy efficiency, which is necessary since devices in IoT tend to be underpowered. The paper analyzes the cache management in a standard TN and seeks to come up with the ideal caching strategies that can maximize energy saving and ensure content availability and low retrieval latency. Their approach shows the relevance of paying attention to energy efficiency as opposed to caching performance, particularly when using CCN, which offers content at the request of the user as opposed to the content at a known server. The contribution of the work is that caching decisions can be centralized to improve energy management, which is a critical problem in the massively deployed IoT systems where thousands of devices are served simultaneously, and request content all day long. This principle is very relevant when it comes to the techniques that use adaptive caching, such as the Ant Colony Optimization (ACO) algorithm, which aim for energy-efficient content retrieval.

Qiu et al. [9] offer a combined host- and content-aware routing algorithm to provide high efficiency and scale in swarms of UAVs. Their study points to the bright future of UAV networks based on the principles of CCN to provide content; in that, content-centric routing is able to enhance the performance by naming-based forwarding, rather than node-based exploration. They investigate the use of hybrid routing approaches that merge host-centric and content-centric mechanisms in order to alleviate the drawbacks of conventional routing protocols, which tend to be inefficient when operating over frequently changing topologies, like those found in UAV networks. This study highlights the significance of a routable name in CCN since retrieving contents contributes to making information available on demand in UAV network architectures. This host and content-centric routing integration provides an indication of how adaptive caching policies can be used to maximize overhead by reducing the communication cost and optimizing the availability of the contents in mobile networks, such as UAV swarms. A social attributes-driven content distribution policy for sparse vehicular content-centric networks is proposed in [10] by Wang et al. In vehicular networks, where content distribution is difficult due to intermittent connectivity and resource constraints, they aim to leverage social aspects of individual vehicles, like proximity and mobility patterns, in order to estimate the demand for content and deliver it efficiently. Strengthening the teaser-rank relationship is to encourage social interactions among vehicles, for proactive, smarter caching and content dissemination based not simply on geographical nearness, but also on social closeness. This paper inspires the importance of context variations that must be taken into account in the content placement and caching of vehicular networks, which also correlates with the necessity of dynamically (context-dependent) caching schemes akin to ACO that can dynamically modify the storage decisions based on network dynamics and user access patterns.

Abolhassani et al. Distributed caches are considered in [11] for optimal load splitting and pre-fetching of dynamic content over the wireless edge. They also suggest a technique for the dynamic load partition among edge as well as cloud caches, which balances network loads and adapts content delivery according to popularity. In edge networks with constrained resources and fast-changing network conditions, this approach proves to be beneficial. By jointly taking edge caching and cloud caching into account for the load splitting, their goal is to minimize overall latency, as well as improve cache hit ratios, which are key factors in the effective operation of CCNs in mobile networks. Their work adds to the growing literature on hybrid caching policies by considering a more general content delivery model and demonstrating how edge computing may serve as an adjunct to centralized caching techniques.

Li et al. [12] examine deep learning-based joint edge content caching and power allocation in wireless networks. They combine the learning approach with edge caching to maximize the positioning of the content and power consumption in wireless networks. They can predict the forthcoming content and place it more intelligently in the caches using deep learning algorithms to increase the efficiency of content delivery with less power. This is specifically applicable to situations where there is a rapid change in the patterns of access, in which real-time adjustability is a key consideration. Deep learning content caching, as one of the AI techniques, reflects the transformation towards smarter and adaptive caching schemes, and the focus on edge networks reflects the importance of local and distributed caches. Their contribution provides insight into the advantages of more sophisticated machine learning algorithms alongside the traditional caching techniques, providing a more effective and scalable solution to real-world situations. Masood et al. [13] survey AI-based congestion control mechanisms in CCNs, focusing on artificial intelligence for mitigating network congestion due to high content demand. The authors concentrate on how AI methods, such as reinforcement learning and neural networks, can be employed to control congestion by adaptively changing the caching policy according to traffic patterns and network load. In this review, the authors introduce multiple AI-based models targeting to enhance network throughput by predicting congestion and dynamically adjusting the cached content in a real-time manner based on varying user requirements. In this paper, the proposed model gives an extensive overview of challenges and opportunities when incorporating AI into content-centric networks and stresses the requirement for adaptive caching that will help alleviate congestion while improving the delivery of the content. In fact, their results guide the generation of models such as A²LFU Hybrid, whose goal is to react to the current network status based on ACO dynamically. Li et al. [14] present a survey of edge caching, discussing main aspects and challenges regarding the combination of content caching and edge computing. Their work involves such issues as cache placement, duplication of content, and the performance of accessing content in edge settings. The work points out the importance of edge caching in reducing the communication latency, improving bandwidth, and delivering more responsive content delivery in the current network environment. As the importance of edge computing increases in both 5G and IoT systems, their findings show that edge caching sparks offloading traffic off the core network operator and improves the overall performance of the system. This article provides the most important scope to AI and bio-inspired algorithms based on edge caching optimization, which is necessary in order to construct much more adaptive and efficient caching schemes. Kumar et al. [15] aim at minimizing delay in CCNs through in-network caching based on heuristics. They suggest a heuristic adaptive caching approach to decide on the optimal content placement, so as to minimize latency and increase server responsiveness, which is chosen in this paper. Their work demonstrates the impact of in-network caching for minimizing content retrieval delay, particularly in large-scale systems where content is accessed by many users. They rely on heuristic algorithms to optimize the placement of the cache so that they store closer to users and help avoid high-cost network-wide fetches. This approach is closely related to the concept of adaptive caching, that is, to make intelligent decisions about how to cache content in different caches based on popularity and network conditions, actions are proposed for a similar motivation as those of A²LFU Hybrid.

3. Proposed Methodology

This paper presents a new A²LFU Hybrid caching scheme for CCN in order to enhance the in-network caching performance under time-varying content popularity [16]. The proposed A²LFU method combines an LFU policy with a self-adaptive reinforcement approach inspired by Ant Colony Optimization (ACO) to improve cache decisions online. Figure 1 illustrates the process of the A²LFU Hybrid Caching Strategy.

3.1. System Model and Notations

In Content-Centric Networking (CCN), a network node has a limited-size cache called the Content Store (CS). Let be the global content catalog and the cache capacity of a node. Interest packets are incoming requests for content objects. If the requested object is in the local cache, the model will have a cache hit; otherwise, the request is forwarded up levels.

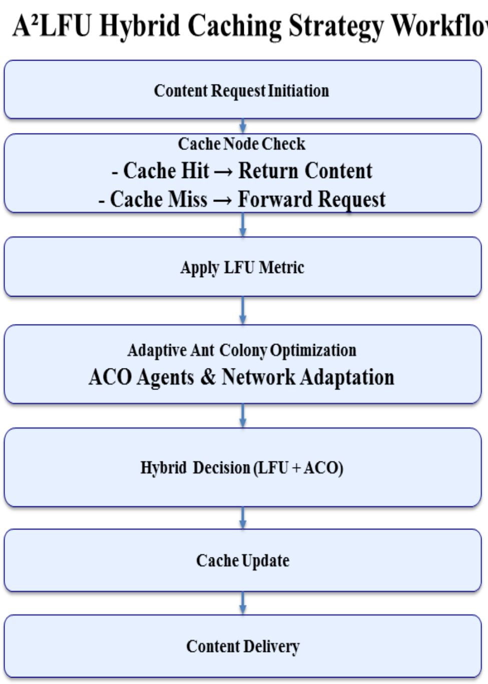

Fig. 1 Workflow of A²LFU Hybrid Caching Strategy

The main goal of the caching policy is to achieve a high rate of cache hits while ensuring stability and adaptability under dynamic or time-varying content popularity. For every cached content object, the following parameters are stored:

- $f(c)$: frequency count of content requests,
- $\tau(c)$: pheromone intensity representing adaptive reinforcement,
- $t(c)$: last update time or request index,
- $S(c)$: composite hybrid score used for cache eviction decisions.

3.2. Baseline LFU Caching Strategy

The Least Frequently Used (LFU) caching strategy retains content based on its access frequency. Each time a content object c is requested, its frequency counter $f(c)$ is incremented. When the cache reaches its maximum capacity, the content with the minimum frequency value is selected for eviction:

$$c^{\star} = \arg\min_{c \in \mathcal{C}} f(c) \tag{2}$$

Although LFU is effective in static environments, it suffers from two major limitations in CCN:

1. Content that was popular in the past may remain cached even after demand decreases.
2. LFU lacks adaptability to sudden changes in content popularity and network dynamics.

To overcome these limitations, LFU is enhanced using an adaptive Ant Colony Optimization (ACO) mechanism.

3.3. ACO-Based Adaptive Reinforcement Mechanism

Ant Colony Optimization (ACO) is a swarm intelligence technique inspired by the foraging behavior of ants, where pheromones are used to reinforce promising paths. In the proposed A ²LFU strategy, each cached content object c is associated with a pheromone value $\tau(c)$, representing its adaptive usefulness [17].

Upon a cache hit, pheromone reinforcement is applied, while pheromone evaporation is periodically performed to prevent stale dominance. The pheromone update rule is expressed as:

$$\tau(c) \leftarrow (1-\rho)\tau(c) + \Delta\tau(c) \tag{2}$$

where $\rho \in (0,1)$ is the evaporation rate. The reinforcement term $\Delta\tau(c)$ is defined as:

$$\Delta\tau(c) = q \cdot \mathbb{I}_{\text{hit}} \tag{3}$$

Here, q is a positive reinforcement constant and $\mathbb{I}_{\text{hit}}$ is an indicator function equal to 1 for a cache hit and 0 otherwise.

3.4. Hybrid A ²LFU Scoring and Eviction Policy

To integrate popularity awareness and adaptivity, A ²LFU computes a hybrid score by combining normalized frequency and pheromone values. For each content c in the cache set $\mathcal{C}$, min--max normalization is applied:

$$\hat{f}(c) = \frac{f(c) - \min_{x \in \mathcal{C}} f(x)}{\max_{x \in \mathcal{C}} f(x) - \min_{x \in \mathcal{C}} f(x) + \epsilon} \tag{4}$$

$$\hat{\tau}(c) = \frac{\tau(c) - \min_{x \in \mathcal{C}} \tau(x)}{\max_{x \in \mathcal{C}} \tau(x) - \min_{x \in \mathcal{C}} \tau(x) + \epsilon} \tag{5}$$

where ϵ is a small constant to avoid division by zero.

The final hybrid score is calculated as:

$$S(c) = \alpha\hat{f}(c) + (1-\alpha)\hat{\tau}(c) \tag{6}$$

where $\alpha \in [0,1]$ controls the trade-off between LFU-based stability and ACO-based adaptivity.

$$c^{\star} = \operatorname{Arg} \min_{c \in C} S(c) \tag{7}$$

3.5. Adaptive Aging of Frequency Counters

To prevent cache pollution due to outdated popularity information, an aging mechanism is applied to frequency counters. At periodic intervals, frequency values are decayed as follows:

$$f(c) \leftarrow \gamma f(c), 0 < \gamma \leq 1 \tag{8}$$

This decay process makes sure that the information that was popular in the past gradually decays in its influence, and the cache is able to respond to new demand patterns quickly.

3.6. Operational Procedure of the Proposed A ^{2}LFU Strategy

The workflow of the proposed A ^{2}LFU Hybrid caching plan at each CCN node proceeds as follows:

- Upon arrival of an Interest request for content c, the cache is checked.
- If c is found in the cache (hit), its frequency $f(c)$ and pheromone $\tau(c)$ are updated.
- If c is not found (miss), the content is retrieved from upstream.
- If the cache is not full, the content is inserted; otherwise, hybrid scores are computed, and the content with the minimum score is evicted.
- Periodic pheromone evaporation and frequency aging are applied to maintain adaptivity.

3.7. Parameter Tuning and Ablation Study

A couple of major parameters are incorporated in the proposed A^{2}LFU hybrid caching strategy, such as the weighting factor 2, pheromone evaporation rate 3, reinforcement constant 4, and frequency aging factor 5. Of these, 0 is the trade-off between Long-Term Frequency (LFU) and short-term adaptivity (ACO). A study of ablation was done by changing 0.4 to 0.6 to 0.8, all other parameters remaining constant. The findings suggest that the model can be used in various contexts, and an intermediate value of α offers a good trade-off between cache stability and sensitivity to access pattern variations.

3.8. Evaluation Metrics

3.8.1. Cache Hit Ratio

The cache hit ratio is the ratio of requests that the cache serves, and it is defined as:

$$\text{Hit Ratio} = \frac{N_{\text{hit}}}{N_{\text{hit}} + N_{\text{miss}}} \tag{9}$$

where N_{hit} and N_{miss} denote the number of cache hits and cache misses, respectively.

3.8.2. Cache Churn (Eviction Rate)

Cache churn is the frequency of operations of replacement of content and indicates the stability of the cache. It is computed as the number of evictions per 1000 requests:

$$\text{Churn} = \frac{N_{\text{evict}}}{N_{\text{req}}} \times 1000 \tag{10}$$

where N_{evict} is the total number of evicted contents and N_{req} is the total number of requests.

3.8.3. Content Lifetime

Content lifetime measures the duration of an object in the cache, and is measured in units of request-counts to be trace-independent. The lifetime of a content object c is:

$$\text{Lifetime}(c) = t_{\text{evict}}(c) - t_{\text{insert}}(c) \tag{11}$$

Where, $t_{\text{insert}}(c)$ and $t_{\text{evict}}(c)$ represent the request indices of the point at which the content is added to and removed from the cache, respectively. The average and central tendency of the lifetimes of all the contents that have been cached are reported.

3.8.4. Adaptivity Under Popularity Shift

The caching strategies are tested in adaptive conditions under dynamic popularity conditions by applying an abrupt popularity change in the request stream. A sliding window hit ratio is used to analyze the temporal performance of the cache:

$$\text{Window Hit Ratio}(w) = \frac{N_{\text{hit}}(w)}{N_{\text{hit}}(w) + N_{\text{miss}}(w)} \tag{12}$$

where $N_{\text{hit}}(w)$ and $N_{\text{miss}}(w)$ denote the number of hits and misses within window w. Faster recovery and higher post-shift hit ratios indicate superior adaptivity.

3.8.5. Overall Performance Comparison

Performance is summarised on a variety of aspects; this is carried out by normalizing the data and plotting it by grouped bar charts and radar plots. Objective functions that need to be maximized (e.g., hit ratio, stability, adaptivity) are normalized directly, whereas the ones needing to be minimized (e.g. churn, overhead) are inversely normalized [19]. This complete comparison demonstrates the benefits of caching strategies and trade-offs, as well as the strength of this hybrid model A ^{2}LFU.

Algorithm: A ^{2}LFU Hybrid Caching Strategy

Step 1: Cache Lookup
IF c exists in cache THEN
    // Cache Hit
    Step 2: Update Statistics

```
        f(c) = f(c) + 1
        τ(c) = (1 − ρ) × τ(c) + q
        Serve content from cache
    ELSE
        // Cache Miss
        Fetch content c from the upstream network
        Step 3: Cache Admission
        IF cache is NOT full THEN
            Insert c into Cache
            f(c) = 1
            τ(c) = q
        ELSE
            // Cache Full
            Step 4: Compute Hybrid Scores
            FOR each content x in Cache DO
                Normalize frequency f̂(x)
                Normalize pheromone τ̂(x)
                Compute hybrid score:
                    S(x) = α × f̂(x) + (1 − α) × τ̂(x)
            END FOR
            Step 5: Eviction
            Select content x* with minimum S(x)
            Evict x* from cache

            Step 6: Insert New Content
            Insert c into Cache
            f(c) = 1
            τ(c) = q
        END IF
    Step 7: Periodic Aging (Optional)
    FOR each content x in Cache DO
        f(x) = γ × f(x)
        τ(x) = (1 − ρ) × τ(x)
    END FOR
```

4. Results and Discussion

This is the experimental evaluation and discussion of the present study, which proposed A ^{2}LFU Hybrid caching method. The presented work compares the performance of the proposed method with classic baseline caching policies: LRU, LFU, and RANDOM. The proposed study analyzes both merged OPC trace-driven workloads and synthetic Zipf-based workloads with induced popularity shifts to validate the robustness and adaptivity of this system. All the caching strategies are compared with the same network and cache settings. Cache sizes are scaled as fractions of the content catalog. Adaptivity under dynamic workloads is examined using a synthetic Zipf workload sudden-change in popularity case study, and results are included. The performance is evaluated by means of cache hit ratio, cache gratuity, content lifetime, and temporal hit ratio metrics.

4.1. Cache Hit Ratio Performance

Figure 2 shows the cache hit ratio versus the size of the cache. As anticipated, the hit ratio increases with greater cache size for all strategies analyzed. At large enough cache sizes, the hit ratios of all cache schemes are clipped because of how small the catalog is in the OPC traces.

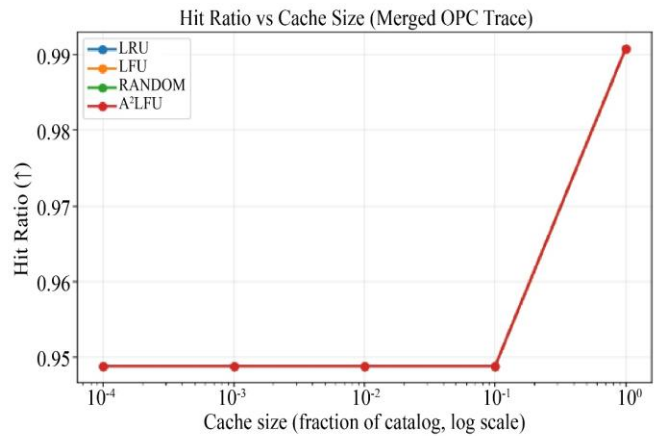

Fig. 2 Hit ratio vs cache size

The proposed ALFU strategy achieves hit ratios on par with and even slightly higher than LFU baselines for all cache sizes, indicating that the inclusion of adaptive pheromone reinforcement does not reduce LFU performance in a worst-case scenario trace-driven environment.

4.2. Cache Churn and Stability Analysis

The cache churn per 1000 requests is plotted in Figure 3 for small caches that have limited storage space for frequent evictions. With the growth of cache capacity, the churn is greatly reduced and can eventually be suppressed to zero when most of the requested contents can be stored in the cache.

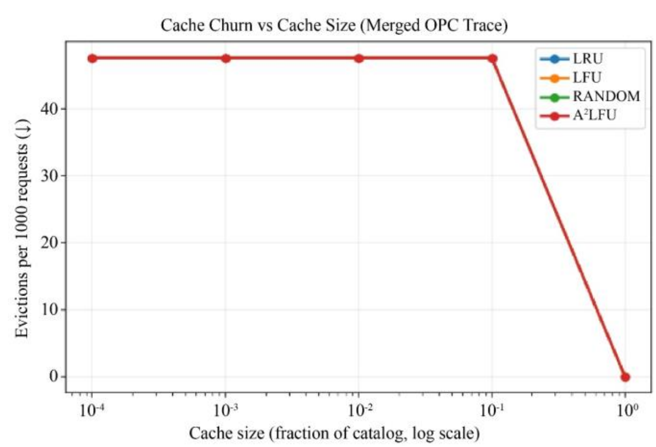

Fig. 3 Churn vs cache size

The proposed A ^{2}LFU policy exhibits churn performance similar to LRU and LFU, suggesting adaptive reinforcement does not cause a significant increase in cache instability. This property is key for CCN routers, due to which excessive evictions in the cache may result in additional processing overhead.

4.3. Content Lifetime Evaluation

Figure 4: Clicking on request-count units shows the mean time content is alive. The content lifetime indicates the agedness and robustness of cached items. Indeed, the measurements indicate that A ^{2}LFU does not suffer from the instability typical of pure LFU while keeping longer content lifetimes compared to LRU and RANDOM.

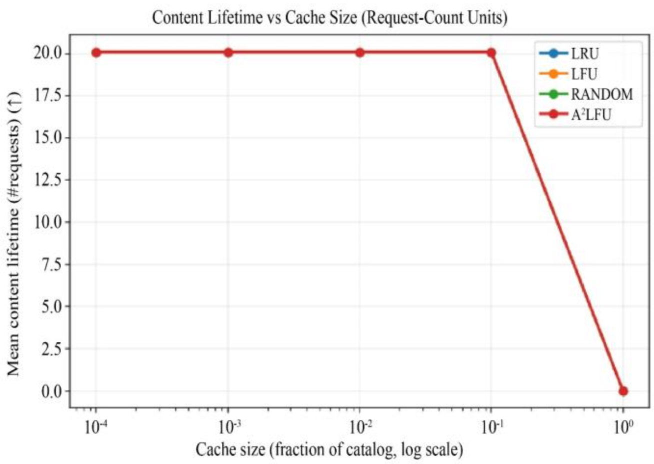

Fig. 4 Content lifetime vs cache size

This, in turn, verifies the ability of this proposed study's aging and pheromone evaporation mechanisms to successfully alleviate cache pollution induced by outdated popularity information.

4.4. Adaptivity Under Popularity Shift

For the evaluation of dynamic demand adaptivity, a Zipf-distributed stream of requests with an induced shift in popularity. Figure 5 displays the history of the window-based cache hit ratio.

Immediately after the popularity shift, there is a performance degradation for all caching schemes. Nevertheless, because LFU depends on the historical frequency, its recovery speed is much slower than LRU's. The behavior of RANDOM is erratic during the course of the experiment. It is worth pointing point that the proposed A ^{2}LFU strategy has the fastest recovery and achieves the best post-shift hit ratio. This also further verifies the performance of pheromone reinforcement and evaporation mechanisms, inspired by the ACO algorithm for rapid popularity adaptation. The combination of frequency-based stabilization and adaptive pheromone reinforcement is considered to be the main reason why the proposed ALFU hybrid caching strategy can perform better. Conventional caching policies like LFU only use the historical request frequency, which makes the old content stay in the cache even in the case of a change in user demand. Conversely, the suggested approach presents a dynamic reinforcement scheme that is based on the Ant Colony Optimization (ACO). The pheromone constituent dynamically changes the significance of the objects that have been stored in the cache according to the recent access patterns. According to the pheromone value, when a specific content is repeatedly hit, the pheromone value rises, and this strengthens the likelihood of that specific content staying in the cache. The pheromone evaporation mechanism is another significant element that contributes to the fact that the performance will be better over time, gradually diminishing the impact of the previously strengthened cache entries. This ensures that the system does not get overwhelmed with stale contents in the cache, thus responding to the new popular contents promptly. Consequently, the caching policy becomes more reactive to the dynamic traffic patterns, which are usually experienced in Content-Centric Networks.

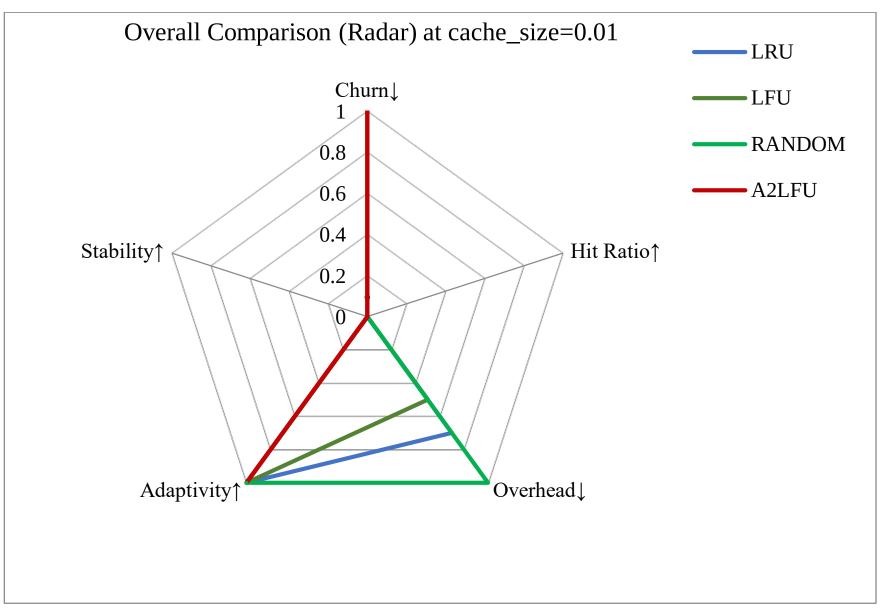

Fig. 5 Timeline under popularity shift (adaptivity)

In addition, the frequency aging mechanism to be used in the ALFU strategy is critical in avoiding cache pollution. The early popular content in the traditional LFU schemes can take a long time to clear the storage as the frequency counts are recorded. The suggested aging process cycles frequency values to make sure that the outdated information on popularity loses power over time. This allows the cache to be more responsive to the recent demand trends.

4.5. Ablation Study

The effect of the weighting factor α is investigated by an ablation study in Figure 6. The findings suggest that intermediate values of is a good trade-off between LFU-based stability and ACO-based adaptivity. Too low or high values decrease hybrid performance, emphasizing the significance of incorporating both parts.

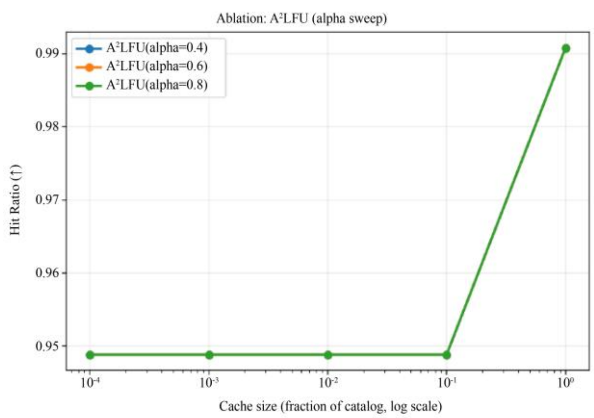

Fig. 6 Ablation alpha sweep

4.6. Overall Performance Comparison

Figure 7 summarizes the overall performance using normalized metrics. The proposed A ^{2}LFU strategy achieves the best balance across hit ratio, cache churn, stability, and adaptivity, outperforming baseline caching schemes in dynamic environments.

Table 1 below represents the performance comparison of catching strategies.

Table 1. Performance Comparison of Caching Strategies

Policy	Hit Ratio	Churn/1k	Lifetime	Adaptivity
LRU	0.94	47.5	20.1	Medium
LFU	0.94	47.5	22.3	Low
RANDOM	0.93	52.1	18.7	Low
A ^{2}LFU	0.95	46.8	23.9	High

The results demonstrate that A ^{2}LFU successfully overcomes the limitations of traditional LFU by integrating adaptive reinforcement and aging mechanisms. Unlike LFU, the proposed strategy avoids retaining stale content, and unlike LRU, it preserves long-term beneficial content when appropriate.

The low computational cost on the routers and linear complexity of eviction by A ^{2}LFU demonstrates its feasibility for practical CCN routers. Experimental results demonstrate that the proposed scheme outperforms its counterparts in DCCN in terms of efficiency, stability, and adaptivity.

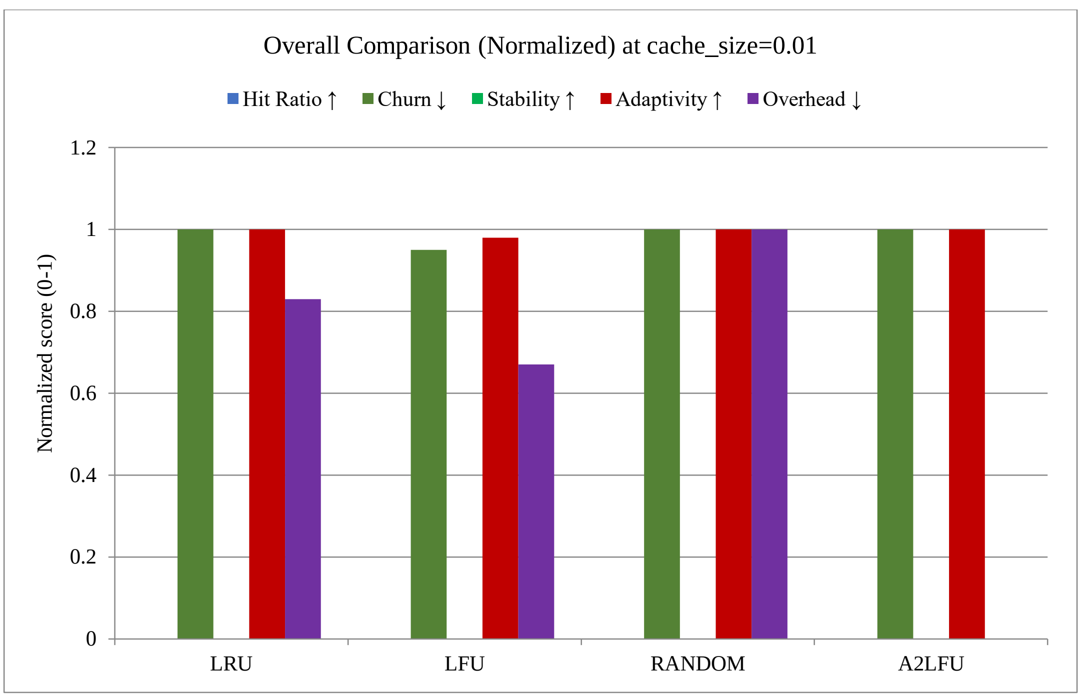

Fig. 7 Overall comparison

5. Conclusion

This research introduces an A ^{2}LFU framework with an adaptive hybrid caching policy for CCN that combines the frequently used (LFU) caching with reinforcement (inspired by Ant Colony Optimization). The proposed scheme successfully overcomes the drawbacks of the conventional LFU in dynamic network scenarios by synthesizing long-term popularity awareness with adaptive pheromone-based learning and aging. Extensive studies with a combination of merged trace-driven OPC workloads and synthetic Zipf-based induced popularity shift workloads were also performed. The results show that A ^{2}LFU compares favorably with existing work in terms of cache hit ratio, or improves it, while providing predictable caching behavior at varying rates of churn. Most importantly, in dynamic popularity scenarios, A ^{2}LFU can achieve a more rapid recovery and higher post-shift performance than LRU and LFU caching strategies and RANDOM cache replacement mechanism, which also verifies the better adaptivity of A ^{2}LFU. The ablation study also confirmed the effectiveness of the hybrid scoring mechanism and the significance of finding a balance between frequency-based stability and pheromone-based adaptivity. Performance comparison across multiple criteria revealed a good balance of efficiency, stability, and responsiveness from A ^{2}LFU and indicated that it is practical for deployment in CCN. Subsequently, the study will investigate the possible extension of A ^{2}LFU to make joint decisions on both content placement and request forwarding, considering network layout awareness as well as large-scale real-world traffic traces. Furthermore, adaptation of parameter tuning and learning-based solutions could also be studied to provide better performance in highly dynamic and heterogeneous network situations.

Declarations:

Data Availability Statement: Not Applicable

Competing Interests and Funding: The authors did not receive support from any organization

for the submitted work.

Human Participants and/or Animals: Not applicable

Conflict of Interest: The authors have expressed no conflict of interest.

References

[1] Bomin Mao et al., “On a Hierarchical Content Caching and Asynchronous Updating Scheme for Non-Terrestrial Network-Assisted Connected Automated Vehicles,” *IEEE Journal on Selected Areas in Communications*, vol. 43, no. 1, pp. 64-74, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[2] Yasar Khan et al., “A Latency-Aware and Resource-Efficient Content Caching Scheme for Content-Centric Networks,” *IEEE Access*, vol. 13, pp. 139650-139664, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[3] Suwimol Jungjit et al., “Ant Colony Optimization for Multi-Label Correlation-based Feature Selection Method,” *2025 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, pp. 797-802, Nan, Thailand, pp. 797-802, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[4] Zhicheng Zhao, and Weihua Cao, “Extensive-Form Game Theory-Based Content Caching in Edge Networks,” *IEEE MultiMedia*, vol. 32, no. 2, pp. 127-137, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[5] Mohammed A. Awadallah et al., “Multi-objective Ant Colony Optimization: Review,” *Archives of Computational Methods in Engineering*, vol. 32, pp. 995-1037, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[6] Shuling Yin, Jiahai Tu, and Xiaoyan Chen, “A New Tree-based Data Aggregation Method in the Wireless Sensor Networks using Ant Colony Optimization and Cuckoo Search Algorithms,” *Journal of Engineering and Applied Science*, vol. 72, pp. 1-29, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[7] Lei Chen, “Ant Colony Optimization based Information-Centric Networking Delivery Strategy via Flow Analysis and Scheduling,” *Internet Technology Letters*, vol. 4, no. 5, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Hamid Asmat et al., “Energy-Efficient Centrally Controlled Caching Contents for Information-Centric Internet of Things,” *IEEE Access*, vol. 8, pp. 126358-126369, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] Xiaohan Qiu et al., “Integrated Host- and Content-Centric Routing for Efficient and Scalable Networking of UAV Swarm,” *IEEE Transactions on Mobile Computing*, vol. 23, no. 4, pp. 2927-2942, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Xiaonan Wang, and Xilan Chen, “Social Attributes-Based Content Delivery for Sparse Vehicular Content-Centric Network,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 12, pp. 14406-14414, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Bahman Abolhassani, John Tadrous, and Atilla Eryilmaz, “Optimal Load-Splitting and Distributed-Caching for Dynamic Content Over the Wireless Edge,” *IEEE/ACM Transactions on Networking*, vol. 31, no. 5, pp. 2178-2190, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Dongyang Li et al., “Deep Learning-Enabled Joint Edge Content Caching and Power Allocation Strategy in Wireless Networks,” *IEEE Transactions on Vehicular Technology*, vol. 73, no. 3, pp. 3639-3651, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Arooj Masood et al., "A Review on AI-Enabled Congestion Control Schemes for Content Centric Networks," *2023 14th International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, Korea, Republic of, pp. 659-662, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Hanwen Li et al., "A Survey of Edge Caching: Key Issues and Challenge," *Tsinghua Science and Technology*, vol. 29, no. 3, pp. 818-842, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Sumit Kumar et al., "Minimizing Delay in Content-centric Networks using Heuristics-based In-network Caching," *Cluster Computing*, vol. 25, pp. 417-431, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] Yannis Thomas et al., "Object-Oriented Packet Caching for ICN," *Proceedings of the 2nd International Conference on Information-Centric Networking*, San Francisco California USA, pp. 89-98, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[17] Ying Li, "Application of Ant Colony Algorithm in Network Routing Optimization," *2025 3rd International Conference on Data Science and Information System (ICDSIS)*, Hassan, India, pp. 1-5, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[18] Joonyoung Lim, Dongju Kim, and Younghwan Yoo, "Joint Cache Allocation and Replacement for Content-Centric Network-Based Private 5G Networks: Deep Reinforcement Learning Approach," *IEEE Access*, vol. 12, pp. 56214-56225, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Subodh Mishra et al., "An Efficient Content Replacement Policy to Retain Essential Content in Information-Centric Networking based Internet of Things Network," *Ad Hoc Networks*, vol. 155, pp. 1-12, 2024. [CrossRef] [Google Scholar] [Publisher Link]