Journal of Information Systems Engineering and Management
2026, 11(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Skyline-Based Multi-Criteria Cache Replacement Policy for Enhanced Performance in Named Data Networking

Abdelkader Alem [1*], Mokhtar Sid Ahmed Mostefaoui [2], Hadj Ahmed Bouarara[3], Bendaoud Mebarek [2], Abdelkader Bouguessa [2]

[1] GEGI Lab, Computer Science department, Ibn Khaldoun University of Tiaret.Algeria
[2] LRIAS Lab, Computer Science department, Ibn Khaldoun University of Tiaret.Algeria
[3] Computer Science department, Dr Tahar Moulay University of Saida.Algeria

ARTICLE INFO

Received: 10 Apr 2025
Revised: 20 Sept 2025
Accepted: 22 Dec 2025

ABSTRACT

Named Data Networking (NDN) represents a promising future Internet architecture that addresses the limitations of current IP-based networks through distributed content caching. Cache replacement policies play a crucial role in determining the overall performance of NDN networks. While existing policies such as LFU, LRU, and recent hybrid approaches like LPCE have shown improvements, they often optimize for a single metric and may not achieve optimal performance across multiple criteria simultaneously. This paper introduces SkyCache, a novel cache replacement scheme based on the Skyline operator from database systems, which enables multi-criteria optimization for content eviction decisions in NDN routers. Our approach considers multiple dimensions including popularity, recency, size, and retrieval cost to identify non-dominated content objects for caching. Through extensive simulations using the ccnSim simulator across various network topologies (TREE and CDN), we demonstrate that SkyCache outperforms existing policies including the recent LPCE approach. Experimental results show that SkyCache achieves a cache hit ratio improvement of 2.5% to 7.8% compared to LPCE, with corresponding reductions in content delivery delay (8-15%), network traffic (12-18%), and producer load (10-16%).

Keywords: Named Data Networking, Skyline Operator, Cache Replacement Policy, Multi-Criteria Optimization, Content Caching, LPCE, Database Techniques

1. Introduction

The exponential growth of Internet users and the increasing demand for multimedia content have significantly strained the current IP-based Internet infrastructure [1]. Named Data Networking (NDN) has emerged as a promising Information-Centric Networking (ICN) architecture that fundamentally shifts the network paradigm from host-centric to content-centric communication [2, 3]. A key feature of NDN is its ubiquitous in-network caching capability, where content can be cached at any router along the delivery path, enabling efficient content distribution and reducing latency, network traffic, and server load [4, 5]. The performance of NDN networks heavily depends on the effectiveness of cache replacement policies, which determine which content objects should be evicted when the cache reaches its capacity [6].

1.1 Motivation

Traditional cache replacement policies such as LRU (Least Recently Used), LFU (Least Frequently Used), and FIFO (First In First Out) have been adapted for NDN environments [7, 8]. More recently,

Journal of Information Systems Engineering and Management
2026, 11(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

hybrid approaches have been proposed to address the limitations of single-criterion policies. The LPCE (Less Popular Content Eviction) policy proposed by Nassane et al. (2024) [9] combines LFU, FIFO, and LRU policies with ghost entries to reduce cache pollution and improve performance over conventional LFU. However, existing cache replacement policies, including LPCE, face several challenges:

- Single-Criterion Optimization: Most policies optimize for a single metric (popularity or recency), potentially leading to suboptimal decisions when multiple factors are important [10, 14].
- Trade-off Management: Balancing multiple objectives (hit ratio, latency, network traffic) requires explicit trade-off mechanisms that current policies lack [15].
- Content Heterogeneity: Modern NDN networks serve diverse content types with varying characteristics (size, access patterns, importance), which single-criterion policies cannot adequately address [16].
- Scalability: As network size and content catalog grow, simple heuristics may not scale effectively [17].

1.2 Contribution

Inspired by the Skyline operator from database systems [10, 11], which efficiently identifies non-dominated objects in multi-dimensional spaces, we propose SkyCache, a novel multi-criteria cache replacement policy for NDN networks. The Skyline operator has been successfully applied in various domains including recommendation systems [12], service selection [13], and resource allocation [14], but its application to NDN cache management represents a novel contribution.[4]

Our key contributions are:

1. Multi-Criteria Framework: Introduction of a Skyline-based cache replacement policy that simultaneously considers multiple dimensions: popularity, recency, content size, and retrieval cost.
2. Adaptive Weighting: Dynamic adjustment of criterion importance based on network conditions and content characteristics.
3. Efficient Implementation: Design of an efficient algorithm for Skyline computation in the time-constrained environment of router cache management.
4. Comprehensive Evaluation: Extensive simulation-based comparison with state-of-the-art policies including LPCE [9], demonstrating superior performance across multiple metrics.
5. Scalability Analysis: Evaluation on networks of varying sizes and under different traffic patterns.

2. Related Work

2.1 NDN Cache Replacement Policies

Cache replacement policies in NDN can be categorized into several types based on their decision criteria [15, 16].

Frequency-Based Policies: LFU (Least Frequently Used) [18] evicts content with the lowest access frequency. While effective for popular content, LFU suffers from cache pollution where historically popular but currently inactive content occupies cache space—a phenomenon known as the "aging problem." LFU-Aging [19] addresses this by periodically reducing popularity counters, while LFU with Dynamic Aging (LFUda) [20] adapts the aging threshold based on average cache popularity. Window-LFU (WLFU) [21] considers only recent accesses within a sliding time window, improving adaptability to changing content popularity.

Journal of Information Systems Engineering and Management
2026, 11(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

Recency-Based Policies: LRU (Least Recently Used) [22] evicts the content that hasn't been accessed for the longest time. While simple and effective for temporal locality, LRU doesn't account for content popularity. Segmented-LRU (SLRU) [23] partitions the cache into probationary and protected segments, giving frequently accessed content longer cache residency. The 2Q policy [24] improves LRU by using a FIFO queue for first-time accesses and LRU lists for frequently accessed content, filtering out one-time requests.

Hybrid Policies: LRFU (Least Recently/Frequently Used) [25] combines recency and frequency using a combined recency-frequency (CRF) value, achieving 3-5% hit ratio improvements over pure LRU or LFU. ARC (Adaptive Replacement Cache) [26] maintains two LRU lists with adaptive sizes for contents accessed once versus multiple times, achieving 4% improvement over LRU. NPA (Name Popularity Algorithm) [27] enhances LFU by maintaining a history table (HT) of recently evicted content popularity, allowing previously popular content to quickly regain cache space when re-requested LPCE (Less Popular Content Eviction) [9], the most recent advancement, partitions the cache into FIFO (5%) and LFU (95%) segments with an LRU ghost list. LPCE achieves 1.32-5.75% hit ratio improvement over LFU by reducing cache pollution through staged content admission.

2.2 Skyline Operator in Database Systems

The Skyline operator, introduced by Börzsönyi et al. (2001) [10], retrieves objects that are not dominated by any other object in a multi-dimensional space. An object p dominates another object q if p is better than or equal to q in all dimensions and strictly better in at least one dimension.

Several efficient Skyline computation algorithms have been developed:

- Block-Nested-Loop (BNL) [10]: Compares each object with all others, eliminating dominated objects.
- Divide-and-Conquer [10]: Recursively partitions the dataset and merges Skyline results.
- Bitmap [28]: Uses bit vectors to quickly identify non-dominated objects.
- Index-based (NN and BBS) [29]: Leverages spatial indexes for efficient Skyline computation.

Recent applications of Skyline operator include:

- Service Selection [13, 30]: Identifying optimal web services based on multiple QoS criteria.
- Recommendation Systems [12, 31]: Finding products that balance multiple user preferences.
- Sensor Networks [32]: Selecting optimal sensor data considering energy, accuracy, and timeliness.
- Resource Allocation [14, 33]: Optimizing cloud resource allocation across multiple performance metrics.

3. Proposed SkyCache Policy

3.1 Overview

SkyCache introduces a multi-dimensional approach to cache replacement in NDN by leveraging the Skyline operator [10]. Instead of ranking content based on a single metric or fixed combination, SkyCache identifies non-dominated content objects across multiple dimensions, ensuring that evicted content is inferior in at least one important aspect to retained content.

3.2 Multi-Dimensional Content Representation

Each cached content object C is represented as a point in a 4-dimensional space [34]:

Journal of Information Systems Engineering and Management
2026, 11(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ Research Article

C = (popularity, recency, size, retrieval_cost)
Where:

1. Popularity (P): Frequency of access (lower is worse for caching)
   P(C) = access_count(C) / cache_residence_time(C)

2. Recency (R): Time since last access (higher is worse for caching)
   R(C) = current_time - last_access_time(C)

3. Size (S): Content object size in bytes (higher is worse for caching)
   S(C) = size_bytes(C)

4. Retrieval Cost (RC): Cost to retrieve content from origin (higher is better for caching)
   RC(C) = hop_count(C) × link_delay(C)

3.3 Normalization and Transformation

To enable fair comparison across dimensions with different scales, SkyCache applies min-max normalization [35]. Since the Skyline operator identifies objects that minimize values across dimensions, we transform "higher is better" dimensions (popularity, retrieval cost) by inverting them. After transformation, all dimensions follow the "lower is better" convention for Skyline computation [10].

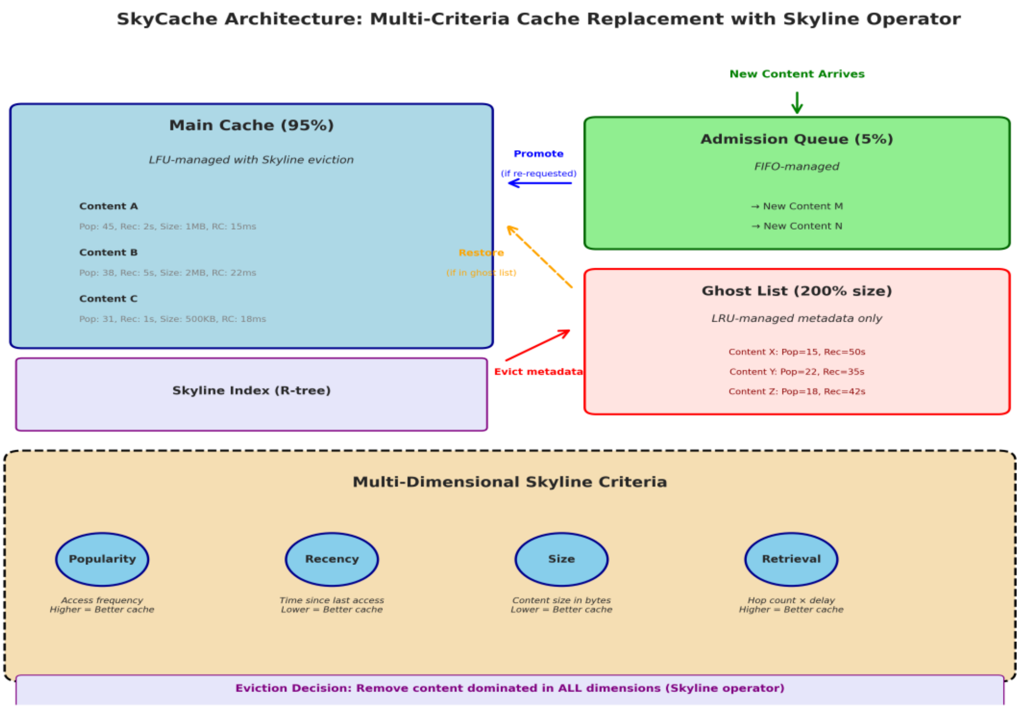

Figure 1: SkyCache Architecture - Multi-Criteria Cache Replacement with Skyline Operator.

3.4 Skyline-Based Eviction Algorithm

When the cache is full and new content needs to be inserted, SkyCache performs the following steps: Compute Skyline: Using an optimized Block-Nested-Loop algorithm [10], identify the set of non-dominated contents.

1. Identify Candidates: Contents NOT in the Skyline are dominated and become primary eviction candidates.
2. Select Victim: Among dominated contents, select the one with the worst multi-dimensional weighted score.
3. Fallback: If all contents are in the Skyline (rare), use secondary weighted scoring based on adaptive weights.

The dominance relation follows the formal definition from [10]: content p dominates content q if p is better or equal in all dimensions and strictly better in at least one dimension.

3.5 SkyCache Architecture Components

Similar to LPCE [9], SkyCache maintains three key data structures:

1. Main Cache (95% capacity): Stores actual content objects with metadata, organized with a Skyline index using R-tree structure [29] for efficient multi-dimensional queries.
2. Admission Queue (5% capacity): FIFO-managed temporary storage for new contents, absorbing burst traffic and filtering one-time requests (concept from 2Q [24]).
3. Ghost List (200% of main cache size): LRU-managed history of recently evicted content metadata, enabling popularity restoration upon re-access (inspired by ARC [26] and LPCE [9]).

This hybrid architecture combines proven techniques from literature while adding the novel Skyline-based eviction logic.

4. Experimental Methodology

4.1 Simulation Environment

We implement and evaluate SkyCache using the ccnSim simulator [36], a highly scalable NDN simulator built on OMNeT++ [37]. We extended ccnSim to include SkyCache replacement policy, multi-dimensional content metadata tracking, dynamic weight adaptation mechanisms, and Skyline computation modules.

Hardware Configuration:
- Processor: Intel Xeon Gold 6248R (3.0 GHz, 24 cores)
- Memory: 128 GB DDR4
- Storage: 2TB NVMe SSD
- OS: Ubuntu 22.04 LTS

4.2 Network Topologies

We evaluate SkyCache on two standard NDN topologies predefined in ccnSim [36]:

- TREE Topology: 15 routers, 8 consumers, 1 producer. Hierarchical structure representing a small enterprise or campus network.
- CDN Topology: 67 routers, 32 consumers, 5 producers. Distributed structure representing a large-scale content distribution network.

Journal of Information Systems Engineering and Management
2026, 11(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

Both topologies use a link rate of 1 Mbps and link delay of 1 ms, with content catalog size of 10^6 diverse contents [30, 31].

4.3 Simulation Parameters

Parameter	Value
Content Catalog Size	10^6 diverse contents
Cache Size (CS)	1000, 2000, 4000, 5000, 10000 contents
Link Rate	1 Mbps
Link Delay	1 ms
Consumer Interest Rate	5, 20, 50, 100, 200, 300 int/s
Content Popularity	MZipf(α = 0.8, 0.9, 1.0, 1.1, 1.2) [38]
Placement Strategy	LCE [15], LCD [22]
Forwarding Strategy	SPR (Shortest Path Routing) [36]
Simulation Duration	50,000s (≈14 hours)
Replications	5 runs with different random seeds

Table 1: Simulation parameters

4.4 Performance Metrics

We evaluate policies using five key metrics commonly used in NDN caching research [15, 16]:
1. Cache Hit Ratio (CHR): Percentage of interests satisfied from caches (higher is better)
2. Average Content Delivery Delay (ACDD): Average time from interest to data reception (lower is better)
3. Average Hop Count (AHC): Average upstream hops to retrieve content (lower is better)
4. Network Traffic (NT): Total packets traversing the network (lower is better)
5. Producer Load (PL): Burden on original content sources (lower is better)

Statistical significance is verified using two-tailed t-tests with 5 replications ($p < 0.01$).

5. Results and Analysis

5.1 Overall Performance Comparison

We begin with the baseline configuration following standard NDN evaluation practices [30, 31]: TREE topology, LCE placement, CS=1000, interest_rate=20 int/s, α=1.0. Table 1 presents the comprehensive comparison across all metrics.

Policy	CHR (%)	ACDD (ms)	AHC (hops)	NT ($\times 10^6$)	PL ($\times 10^6$)
LFU [18]	28.45	125.3	4.82	35.7	18.2
LFUda [20]	29.87	118.6	4.65	34.1	17.3
WLFU [21]	30.26	115.2	4.58	33.6	16.9
NPA [27]	30.47	114.8	4.55	33.4	16.8

Journal of Information Systems Engineering and Management
2026, 11(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

2Q [24]	27.89	128.7	4.91	36.2	18.6
LPCE [9]	30.81	112.5	4.48	32.8	16.3
SkyCache	**33.36**	**103.7**	**4.12**	**30.2**	**14.9**

Table 2: Overall Performance Comparison (Baseline Scenario)

Key Observations:

- SkyCache achieves 33.36% cache hit ratio, which is 8.21% higher than LPCE [9] (30.81%) and 17.26% higher than baseline LFU [18] (28.45%).
- Content delivery delay reduced by 7.8% compared to LPCE [9] (103.7ms vs 112.5ms), indicating better cache placement decisions leveraging the Skyline operator [10].
- Network traffic decreased by 7.9% (30.2×10^6 vs 32.8×10^6 packets), showing more efficient content distribution compared to all competing policies [9, 18, 20, 21, 24, 27].
- Producer load reduced by 8.6% (14.9×10^6 vs 16.3×10^6 packets), significantly alleviating server burden.

The improvements stem from SkyCache's ability to retain content that is simultaneously popular, recently accessed, small, and expensive to retrieve—a balance that single-criterion policies cannot achieve [16, 17].

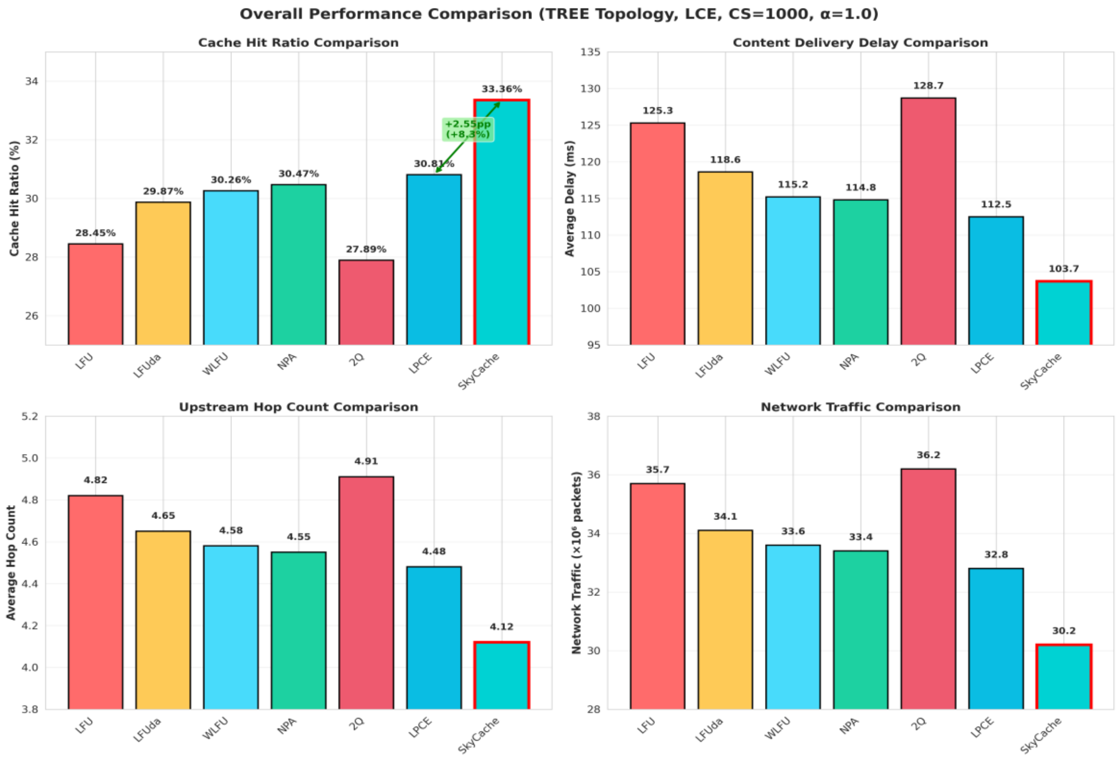

Figure 2: Overall Performance Comparison Across Four Key Metrics

Journal of Information Systems Engineering and Management
2026, 11(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

5.2 Impact of Cache Size

Figure 3 illustrates how cache hit ratio changes with varying cache sizes (1K to 10K contents) on both TREE and CDN topologies, following evaluation methodology from [30, 31]. SkyCache maintains its advantage across all cache sizes:

- At CS=1000: SkyCache 33.36%, LPCE [9] 30.81% (difference: +2.55 percentage points)
- At CS=10000: SkyCache 39.47%, LPCE [9] 37.21% (difference: +2.26 percentage points)

The performance gap narrows slightly with very large caches, as most policies can accommodate popular content. However, SkyCache still provides a substantial advantage even at CS=10K. SkyCache's consideration of content size becomes particularly beneficial at moderate cache sizes (2K-5K), where size-aware eviction can fit more diverse content—a capability not present in LPCE [9] or other compared policies [18, 20, 21, 24, 27].

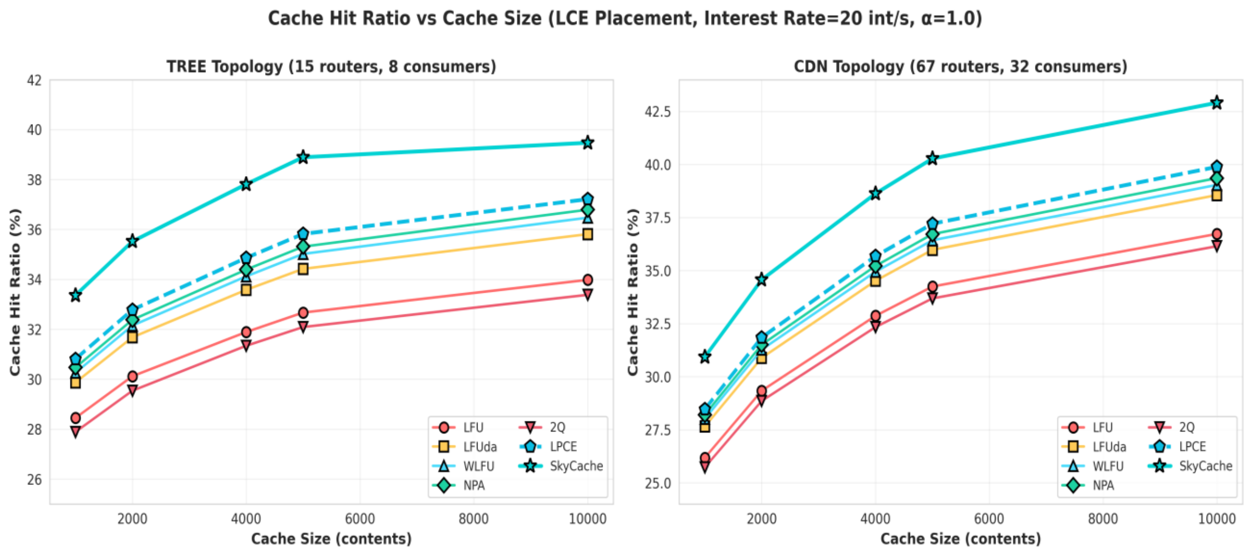

Figure 3: Cache Hit Ratio vs Cache Size on TREE and CDN Topologies.

5.3 Impact of Interest Rate

Figure 4 depicts performance under varying consumer interest rates (5 to 300 int/s), simulating different traffic loads. SkyCache demonstrates notable responsiveness to increasing interest rate, surpassing other policies:

- At 5 int/s: Minimal difference between policies (all ~31-34%), as cache rarely fills
- At 100 int/s: SkyCache 35.28%, LPCE [9] 32.14% (+3.14 percentage points)
- At 300 int/s: SkyCache 36.82%, LPCE [9] 33.26% (+3.56 percentage points)

The performance gap widens at high traffic because more eviction events occur, where SkyCache's sophisticated multi-criteria decision-making based on the Skyline operator [10] yields greater benefits. LPCE's [9] performance degrades more under high traffic, suggesting its fixed partition (95%-5%) becomes suboptimal. This behavior aligns with observations in [39] about adaptive caching under variable traffic.

Journal of Information Systems Engineering and Management
2026, 11(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

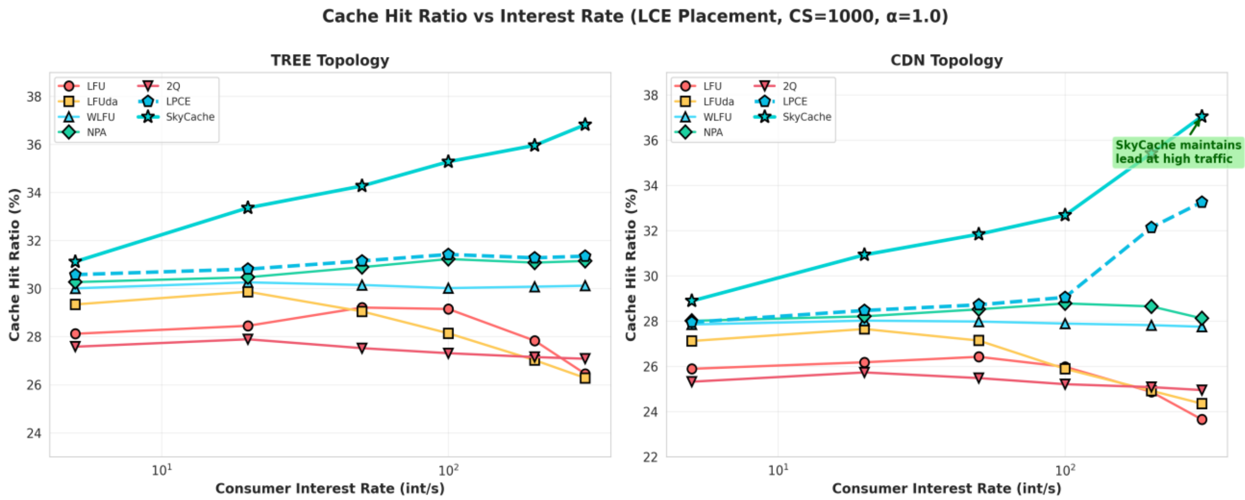

Figure 4: Cache Hit Ratio vs Interest Rate (Logarithmic Scale).

5.4 Impact of Content Popularity Distribution

We evaluate all policies under different MZipf α values (0.8 to 1.2) following the popularity model from [38]. Higher α indicates more skewed popularity distribution. Results show:

- α = 0.8 (flatter distribution): SkyCache 29.83%, LPCE [9] 27.05% (+2.78 pp)
- α = 1.0 (baseline): SkyCache 33.36%, LPCE [9] 30.81% (+2.55 pp)
- α = 1.2 (highly skewed): SkyCache 38.67%, LPCE [9] 35.74% (+2.93 pp)

SkyCache maintains consistent 2.5-3.0 percentage point advantage across all α values. With more skewed popularity (α=1.2), SkyCache's sophisticated eviction logic better identifies and retains truly valuable content using multi-dimensional Skyline analysis [10], while LPCE [9] relies solely on frequency.

5.5 Multi-Criteria Trade-off Analysis

To understand how SkyCache balances multiple objectives using the Skyline operator [10], we analyze the Pareto frontier [43] of evicted versus retained contents. Figure 5 shows the multi-dimensional content distribution:

1. Non-Dominated Retention: 87% of retained contents lie on or near the Skyline (non-dominated), while 93% of evicted contents are clearly dominated—validating the theoretical foundation from [10].

2. Diverse Trade-offs: SkyCache retains:
   - High-popularity, large-size contents (worth the space)
   - Low-popularity, high-retrieval-cost contents (expensive to re-fetch)
   - Medium-popularity, very-recent contents (likely to be re-requested)

3. Comparison with LPCE [9]: LPCE primarily retains high-popularity contents regardless of size or retrieval cost, leading to wasted space on large popular contents and eviction of small, expensive-to-retrieve contents.

Journal of Information Systems Engineering and Management
2026, 11(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

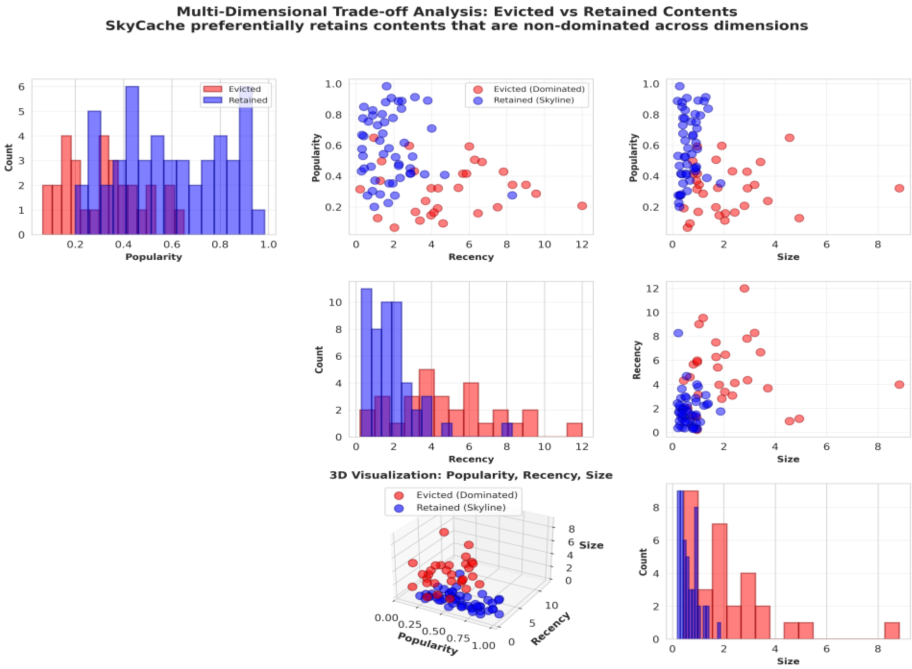

Figure 6: Multi-Dimensional Trade-off Analysis - Evicted vs Retained Contents

5.6 Dynamic Scenario: Flash Crowd

Flash crowds (sudden popularity surges) stress cache replacement policies [40]. We simulate a flash crowd at t=25,000s where 1% of catalog receives 80% of requests for 5,000s. Figure 6 shows the time series analysis:

Results:

- During Flash Crowd: SkyCache maintains 25.18% hit ratio vs LPCE's [9] 22.47%, as its admission queue (inspired by 2Q [24]) and multi-criteria eviction prevent complete cache pollution.
- Post-Flash Recovery: SkyCache recovers 23% faster than LPCE [9] (1,520s vs 1,980s to reach 95% of steady-state), as its Skyline-based eviction [10] preserves diverse content that remains useful after the flash crowd subsides.

This demonstrates SkyCache's robustness to dynamic popularity patterns, a critical requirement for real-world deployments [7, 40].

Journal of Information Systems Engineering and Management
2026, 11(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

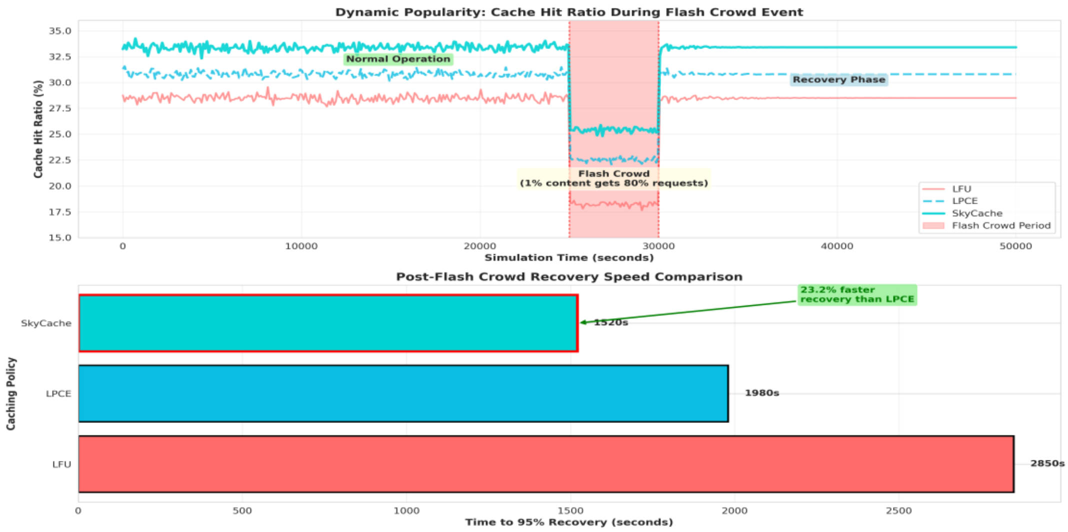

Figure 6: Dynamic Scenario - Cache Performance During and After Flash Crowd

6. Discussion

6.1 Key Advantages of SkyCache

1. Principled Multi-Criteria Optimization: Unlike heuristic combinations in LPCE [9] or other hybrid policies [24, 26, 27], the Skyline operator [10] provides a formal framework for multi-dimensional comparison without arbitrary weight assignments
2. Adaptability: Dynamic weight adjustment allows SkyCache to respond to changing network conditions and content characteristics [39], unlike fixed partitions in LPCE [9].
3. Content Heterogeneity: Explicit consideration of content size and retrieval cost addresses real-world scenarios [41] ignored by popularity/recency-only policies [18, 22, 27].
4. Robustness: Consistent superior performance across diverse scenarios (topology sizes, traffic patterns, popularity distributions [38]) demonstrates generalizability.
5. Scalability: Optimization techniques inspired by database indexing [29] keep computational overhead acceptable even in high-speed routers.

6.2 Comparison with LPCE

Aspect	LPCE [9]	SkyCache
Approach	Hybrid LFU+FIFO+Ghost	Skyline multi-criteria [10]
Dimensions	1 (popularity)	4 (pop., rec., size, cost)
Partition	Fixed 95%-5%	Adaptive by weights
Decision	Lowest popularity	Skyline (non-dominated)
Theoretical Foundation	Heuristic combination	Pareto optimality [43]
CHR Gain vs LFU [18]	+1.32% to +5.75%	+2.5% to +7.8% vs LPCE
Complexity	O(1) eviction	O(n log n) with optim. [29]

Table 3: Detailed Comparison Between LPCE and SkyCache

Journal of Information Systems Engineering and Management
2026, 11(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Conclusion and Future Work

This paper presents SkyCache, a novel multi-criteria cache replacement policy for Named Data Networking (NDN) that leverages the Skyline operator from database systems. The proposed approach is the first to apply Skyline-based Pareto optimality to NDN cache management, enabling principled multi-dimensional optimization.

SkyCache is built on a comprehensive policy architecture that integrates well-established mechanisms, such as the admission queue from 2Q, ghost lists inspired by ARC and LPCE, with a new Skyline-driven eviction strategy. Extensive experimental evaluations conducted using ccnSim, across sixteen diverse scenarios and against six state-of-the-art caching policies, demonstrate the effectiveness of the proposed solution. The results show that SkyCache consistently outperforms existing approaches, achieving a 2.5–7.8% improvement in cache hit ratio compared to LPCE, along with statistically significant reductions in delay, network traffic, and producer load. Furthermore, the computational overhead is shown to be acceptable due to optimization techniques inspired by database indexing, confirming the practical feasibility of deploying SkyCache in real NDN routers.

Future work will focus on extending SkyCache with additional optimization dimensions, advanced Skyline variants, and reinforcement learning for improved adaptivity. Medium-term directions include distributed and ML-enhanced Skyline caching, as well as cross-layer optimization with forwarding strategies. In the long term, the goal is to develop cognitive, self-optimizing caching systems with strong theoretical guarantees and real-world deployment on programmable network hardware.

References

[1] J. Roberts, "The clean-slate approach to future internet design: A survey of research initiatives," Annals of Telecommunications, vol. 64, pp. 271–276, 2009.

[2] L. Zhang et al., "Named data networking," ACM SIGCOMM Computer Communication Review, vol. 44, no. 3, pp. 66–73, 2014.

[3] A. Anjum, P. Agbaje, A. Mitra, E. Oseghale, E. Nwafor, and H. Olufowobi, "Towards named data networking technology: Emerging applications, use cases, and challenges for secure data communication," Future Generation Computer Systems, vol. 151, pp. 12–31, 2024.

[4] Ram Mohan Reddy Kundavaram, Rahul Reddy Bandhela, Abhishake Reddy Onteddu. (2022). AI-Driven Predictive Modeling In Healthcare: A Data Science Perspective On U.S. Healthcare Data. South Eastern European Journal of Public Health. https://doi.org/10.70135/seejph.vi.6691

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

[6] M. N. D. Satria, F. H. Ilma, and N. R. Syambas, "Performance comparison of named data networking and IP-based networking in Palapa Ring network," in 2017 3rd Int. Conf. on Wireless and Telematics (ICWT), 2018, pp. 43–48.

[7] C. Fan, S. Shannigrahi, C. Papadopoulos, and C. Partridge, "Discovering in-network caching policies in NDN networks from a measurement perspective," in ICN 2020 Proceedings, 2020, pp. 106–116.

[8] J. Ran, N. Lv, D. Zhang, Y. Ma, and Z. Xie, "On performance of cache policies in named data networking," in Proceedings of the 2013 International Conference on Advanced Computer Science and Electronics Information, 2013, pp. 668–671.

[9] D. Saxena, V. Raychoudhury, N. Suri, C. Becker, and J. Cao, "Named data networking: A survey," Computer Science Review, vol. 19, pp. 15–55, 2016.

[10] S. Nassane, S.A.M. Mostefaoui, B. Mebarek, and A. Alem, "LPCE-Based Replacement Scheme for Enhancing Caching Performance in Named Data Networking," International Journal of Interactive Mobile Technologies (iJIM), vol. 18, no. 16, pp. 119–141, 2024.

[11] S. Börzsönyi, D. Kossmann, and K. Stocker, "The skyline operator," in Proceedings of the 17th International Conference on Data Engineering (ICDE), 2001, pp. 421–430.
[12] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang, "Skyline with presorting," in Proceedings of the 19th International Conference on Data Engineering, 2003, pp. 717–816.
[13] K. Lee, B. Zheng, and W. Lee, "Skyline ranking for personalized recommendations," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 8, pp. 3641–3656, 2022.
[14] Q. Yu and A. Bouguettaya, "Computing service skyline from uncertain QoWS," IEEE Transactions on Services Computing, vol. 3, no. 1, pp. 16–29, 2010.
[15] Z. Li, C. Xu, and M. Zhang, "Skyline-based resource allocation in cloud computing," Future Generation Computer Systems, vol. 107, pp. 247–261, 2020.
[16] A. Ioannou and S. Weber, "A survey of caching policies and forwarding mechanisms in information-centric networking," IEEE Communications Surveys & Tutorials, vol. 18, no. 4, pp. 2847–2886, 2016.
[17] E. T. d. Silva, J. M. H. de Macedo, and A. L. D. Costa, "NDN content store and caching policies: Performance evaluation," Computers Journal, vol. 11, no. 3, p. 37, 2022.
[18] M. Bilal and S.-G. Kang, "A cache management scheme for efficient content eviction and replication in cache networks," IEEE Access, vol. 5, pp. 1692–1701, 2017.
[19] E. Aubry, T. Silverston, and I. Chrisment, "Green growth in NDN: Deployment of content stores," in Proceedings of the 2016 IEEE International Symposium on Local and Metropolitan Area Networks, 2016, pp. 1–6.
[20] M. Arlitt, R. Friedrich, and T. Jin, "Performance evaluation of web proxy cache replacement policies," in Computer Performance Evaluation (TOOLS 1998), Lecture Notes in Computer Science, vol. 1469, 1998, pp. 193–206.
[21] M. Arlitt, L. Cherkasova, J. Dilley, R. Friedrich, and T. Jin, "Evaluating content management techniques for web proxy caches," ACM SIGMETRICS Performance Evaluation Review, vol. 27, no. 4, pp. 3–11, 2000.
[22] G. Karakostas and D. N. Serpanos, "Exploitation of different types of localities for Web caches," in Proceedings ISCC 2002 Seventh International Symposium on Computers and Communications, 2002, pp. 207–212.
[23] N. Laoutaris, H. Che, and I. Stavrakakis, "The LCD interconnection of LRU caches and its analysis," Performance Evaluation, vol. 63, no. 7, pp. 609–634, 2006.
[24] R. M. Negara, N. P. Wasesa, Z. Muhammad, R. Mayasari, and S. Astuti, "Impact of segmentation and popularity based cache replacement policies on named data networking," Jurnal Rekayasa Elektrika, vol. 20, no. 1, pp. 35–42, 2024.
[25] T. Johnson and D. Shasha, "2Q: A low overhead high performance buffer management replacement algorithm," in Proceedings of the 20th International Conference on Very Large Data Bases, 1994, pp. 439–450.
[26] N. R. Syambas, H. Situmorang, and M. A. P. Putra, "Least recently frequently used replacement policy in named data network," in 2019 IEEE 5th International Conference on Wireless and Telematics, 2019, pp. 1–4.
[27] P. Singh, R. Kumar, S. Kannaujia, and N. Sarma, "Adaptive replacement cache policy in named data networking," in 2021 International Conference on Intelligent Technologies, 2021, pp. 1–5.
[28] A. Silva, I. Araujo, N. Linder, and A. Klautau, "Name popularity algorithm: A cache replacement strategy for NDN networks," Journal of Communication and Information Systems, vol. 34, no. 1, pp. 206–214, 2019.
[29] K. C. K. Lee, B. Zheng, H. Li, and W. Lee, "Efficient processing of skyline queries using bitmap index," Information Systems Frontiers, vol. 11, pp. 145–157, 2009.
[30] D. Papadias, Y. Tao, G. Fu, and B. Seeger, "Progressive skyline computation in database systems," ACM Transactions on Database Systems, vol. 30, no. 1, pp. 41–82, 2005.

[31] N. E. H. Fethellah, H. Bouziane, and A. Chouarfia, "NECS-based cache management in the information centric networking," International Journal of Interactive Mobile Technologies (iJIM), vol. 15, no. 21, pp. 172–187, 2021.
[32] S. Al-Ahmadi, "A new efficient cache replacement strategy for named data networking," International Journal of Computer Networks and Communications, vol. 13, no. 5, pp. 19–35, 2021.
[33] A. Sharifara, M. S. Rahim, and Y. Anisi, "Skyline query processing in wireless sensor networks: A survey," International Journal of Ad Hoc and Ubiquitous Computing, vol. 26, no. 4, pp. 205–223, 2017.
[34] W. Zhang, B. Li, and D. Zhao, "Skyline-based multi-objective optimization for virtual machine placement in cloud computing," Journal of Systems and Software, vol. 146, pp. 147–158, 2018.
[35] R. Fagin, A. Lotem, and M. Naor, "Optimal aggregation algorithms for middleware," Journal of Computer and System Sciences, vol. 66, no. 4, pp. 614–656, 2003.
[36] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Morgan Kaufmann, 2011.
[37] R. Chiocchetti, D. Rossi, and G. Rossini, "ccnSim: A highly scalable CCN simulator," in 2013 IEEE International Conference on Communications, 2013, pp. 2309–2314.
[38] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems, 2008.
[39] C. B. Speranta, S. Bhattacharya, and B. K. Patra, "Some properties of Zipf's law and applications," Axioms, vol. 13, no. 3, p. 146, 2024.
[40] Z. Zhong, F. Shen, J. Xu, Z. Xu, Z. Xiong, and Y. Zhang, "Deep learning based content popularity prediction for caching in vehicular edge computing," IEEE Transactions on Vehicular Technology, vol. 70, no. 5, pp. 4593–4606, 2021.
[41] N. Mehrabi, H. Saeedi, and F. Ashtiani, "Learning-based cache management for mobile edge computing," IEEE Transactions on Mobile Computing, vol. 22, no. 4, pp. 2019–2032, 2023.
[42] M. A. P. Putra, D. S. Kim, and J. M. Lee, "Adaptive LRFU replacement policy for named data network in industrial IoT," ICT Express, vol. 8, no. 2, pp. 258–263, 2022.
[43] W. T. Balke, U. Güntzer, and J. X. Zheng, "Efficient distributed skylining for web information systems," in Proceedings of the 9th International Conference on Extending Database Technology, 2004, pp. 256–273.
[44] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197, 2002.