

# NECS-based Cache Management in the Information Centric Networking

<https://doi.org/10.3991/ijim.v15i21.20011>

Nour El Houda Fethellah<sup>1</sup>, Hafida Bouziane<sup>1,2</sup>, Abdallah Chouarfia<sup>1,2</sup>(✉)

<sup>1</sup>Université des Sciences et de la Technologie d'Oran-Mohamed Boudiaf USTO-MB,  
Oran, Algérie

<sup>2</sup>Université Ibn Khaldoun, Tiaret, Algeria  
abdallah.chouarfia@univ-usto.dz

**Abstract**—Information Centric Networking (ICN) architectures propose to overcome the problem of Internet architecture. The main straight point of such architectures is the in-network caching. The efficiency of the adopted caching strategy influences the ICN performance. It manages the contents in the network and decides where to cache them. The major issue is the strategic selection of the cache routers to store the data on delivery path. This will reduce congestion, optimize the distance between the consumers and required data, improve latency, and alleviate the viral load on the servers. In this paper, we propose a New Efficient Caching Strategy for Named Data Networking (NDN), named NECS. The NDN is the most promising architecture among all the ICN architectures. The NECS strategy reduces the traffic redundancy, eliminates the useless contents replication, and improves the replay time for users, due to the strategic cache routers selection. We carried out extensive experiments to evaluate the NECS performances against other caching strategies. The results of our conducted simulations revealed that the NECS strategy has interesting and convincing performances in many aspects and records high cache-hit ratio.

**Keywords**—information-centric networking ICN, named data networking NDN, caching strategy, future internet architecture, in-network caching, cache management policies

## 1 Introduction

The explosion of Internet usage in everyday life has created a set of new challenges. Every minute, a huge amount of content broadcasted over the Internet. Globally, people are turning more and more to the network for their daily activities such as work, education, research, shopping, and entertainment. Millions of people are online at the same time, which leads to a massive amount of digital activities at the same time and saturates the network. Internet changes the world in completely unexpected ways and the emerging trends demonstrate the inefficiency of the current Internet architecture, showing the need for a new Internet architecture. It proves that the world has already started a new era, where content and technology play a critical role.

According to the Domo website annual statistics for the year 2020 regarding Internet traffic for one minute, the internet users are more than 4.5 billion users. They send 41,666,667 What Sapp messages. They made 1,388,889 video and voice calls. The Zoom application records 208,333 participants attending online meeting. In LinkedIn, 69,444 people apply for jobs. At the Amazon website, 6,659 Amazon packages shipped. On YouTube, 500 hours video viewed [1].

By 2023, Internet users will be nearly two-thirds of the global population. It is expected that Internet users will be about 5.3 billion, which means 66% of the global population with 3.9 billion additional users compared to 2018 [2].

As mentioned before, Internet increasingly used for massive content dissemination while it stills not well developed to respond adequately to this dominant role. This challenge has motivated many research initiatives to adapt the actual Internet architecture to the current requirements. Despite the significant research efforts in this context, there are still many open issues.

The key issue remains the explosive growth of content demand and the dominance of bandwidth-intensive applications, which increases the content size as well as the Internet traffic. Thus, content/information considered as a core Internet architecture. Among the adopted techniques, the Information-Centric Networking (ICN) architecture tries to overcome such constraints by in-network caching. It allows cache routers to store the disseminated content across the network, eliminating redundant requests and decreasing the network traffic [3].

The Named Data Networking (NDN) is the most prominent architecture among the ICN architectures, proposed as a future Internet architecture [4]. Its perspectives are using names to retrieve contents, securing contents instead of securing channels, adopting caching contents, caching mechanism, and NDN transfer stateful while IP transfer is stateless.

The NDN default caching mechanism consists to leaving a copy of the required content on each NDN cache router located in the delivery path, which alleviates the burden on the original content provider. The intermediates NDN cache routers provide content to consumers. The requested contents provided from a near NDN cache router improve the response time.

However, the NDN default caching mechanism suffers from inefficient use of cache router storage, high cache redundancy, and high replacement overhead, which makes the NDN system inefficient. While, the NDN architecture would be more efficient and robust by suggesting an appropriate content caching mechanism.

It is essential to determine which cache router is authorized to store content before deciding which content to store. In our previously published work, we investigated the requirements and the trade-off for making this decision. In the present paper, we propose NECS, a new efficient caching strategy for NDN architecture and present the experiments conducted to prove its efficiency.

The NECS caching approach is as follows: the network divided into sets, using an enhanced clustering algorithm. For each resulting cluster, three caching routers are selected using Multi-Attribute Decision-Making methods (MADM). Only the main cache router performs storing contents.

An extensive simulation analysis has been conducted to prove NECS approach superiority as in-network caching for the NDN architecture compared to the default NDN caching strategy (LCE) and the two other selected caching strategies (Random and Prob (p)).

The NECS caching strategy enables fetching content from near cache routers by minimizing latency and minimizing traffic overhead leading to higher cache-hit ratio and shorter potential stretches (hops).

The NECS caching approach is the well-suited alternative to the actual default caching strategy of the NDN architecture since it improves the network performance and resilience.

In addition to the introduction, the paper structured as follows: Section II discusses the related work. Section III presents NECS caching strategy concept. Section IV reports the simulation parameters and discusses the obtained results. Finally, section V concludes the paper and gives some perspectives about future work.

## 2 Related work

The NDN architecture is the most prominent architecture among all the proposed ICN approaches as future Internet architecture. It enhances the Quality of Service in different terms such as bandwidth, delay, use of resources, congestion, and load server.

To benefit from the advantages of the NDN architecture, effective cache management needed. Several recent studies have focused specifically on finding optimal caching schemes to enhance the overall network performance. However, the investigations in this context are still at an early stage. An efficient caching requires the election of the most advantageous locations on the delivery path to store disseminated content [4]. In this section, we present some important contributions proposed in the literature.

**Leave a copy everywhere:** LCE [5], [6] is the default caching strategy for the NDN architecture. The LCE stores a copy of the requested data packet on each NDN cache router along the delivery path. It aims to reduce the upstream demand.

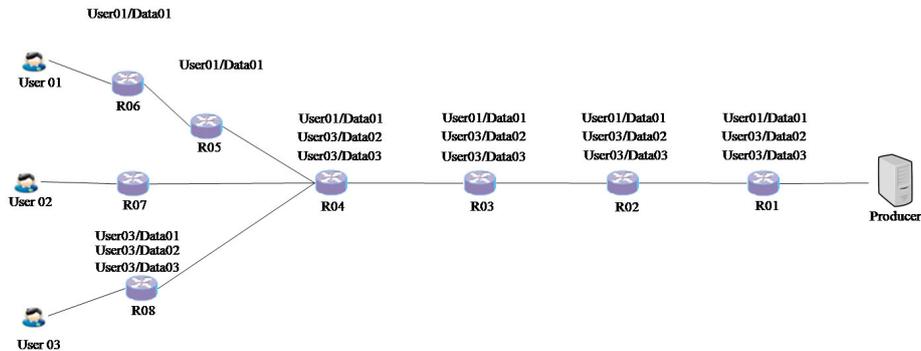


Fig. 1. Illustrative example of LCE caching strategy

Figure 1 presents an illustrative example of the LCE strategy. There are three users : User01, User02, User03, and one Producer. User01 requests Data01. It replicated on all cache routers located between the Producer and User01 (R01, R02, R03, R04, R05, R06). User02 does not request any data. User03 requests Data01, Data02, and Data 03 respectively. User03 requests Data01. It fetches from the cache router R04 and replicates on all the cache routers located along the delivery path. User03 requests Data02. He obtains it from the Producer and it duplicated on all the cache routers on

the delivery path. User03 requests Data03. The Producer satisfies it and replicates it on the delivery path. When User03 requests once more Data01, Data02, or Data03, he satisfied with the cache router R08. The copies on the cache routers R01, R02, and R03 are unnecessary. They only consume the network resources.

The LCE eases the access to content and reduces response time. However, duplicating content on all NDN cache routers along the delivery path wastes network resources. Sometimes, popular content replaced with less popular content due to the limited cache router size.

**Leave a Copy Down: LCD** [7] is a cache management policy. The LCD strategy consists to leaving a copy of the requested data packet at the NDN cache router one level down toward the consumer after each demand. It aims to reduce the cache redundancy in the network. However, it wastes the network resources such as the bandwidth, the NDN cache routers storages on the delivery path, and long time to place content near to user.

**Move a Copy Down: MCD** [8] same as the LCD strategy, it moves the content from the actual cache router to the next cache router on the delivery path. This allows freeing more space on the cache router and reducing the content redundancy. However, it increases the content request delay by removing the requested content from the previous NDN cache router and duplicating the requested content on the next router [9]. The repetitive requests for the same content consume the network resources.

**Most Popular Content: MPC** [10] designed to store only popular content. Each NDN cache router computes the number of demands (popularity count) for each request (content name). It saves the request names and their amount in the popularity table (PT). A requested content is popular when its request number is equals or exceeds the popularity threshold. The NDN cache router stores the popular content. It sends a Suggestion message to its neighbors to save the popular content. The neighbors accept or not to cache the popular content. It depends on their local policies. It reaches high cache-hit ratio and reduces the stretch ratio. However, it records high redundancy and low diversity. The content popularity calculation increases the response delay.

**The random strategy** is a cache management policy [11]. It places only one copy of the requested data packet on a single NDN cache router located on the delivery path. The NDN cache router selection is random.

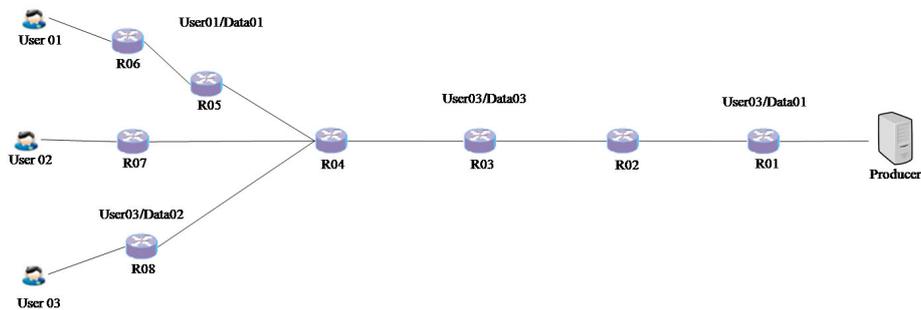


Fig. 2. Illustrative example of Radom caching strategy

Figure 2 presents an illustrative example of the Random strategy. We take the same scenario as the LCE strategy. User01 requests Data01. The Producer satisfies User01 also, a single copy of Data01 is stored on the NDN cache router R05. User03 requests Data01. The NDN cache router R05 does not serve the request. It is not located on the same common part of the delivery path. However, Data01 retrieved from the Producer, and a copy stored on the NDN cache router R01. User03 requests Data02. The Producer serves him, and a copy saved on the NDN cache router R03. User03 requests Data03. It fetched from the Producer and a copy saved on the NDN cache router R08.

The Random strategy is an autonomous randomly cache content policy. It reduces content duplication and increases the content diversity. It achieves high cache hit, reduces the delay, and reaches low overhead. However, the arbitrary location of a copy decreases the NDN architecture effectiveness. It has an unpredictable nature [12].

**The Prob(p) strategy** is a non-cooperative policy. It aims to store the data packet copies on NDN cache routers located on the delivery path with a defined probability  $p$  and does not cache the data packet with probability  $(1-p)$ . When the NDN cache router receives the data packet, it generates a random number from zero to one. If the generated number is little than the value  $p$ , the NDN cache router duplicates the requested data packet. Otherwise, the NDN cache router disseminates the requested data packet without storing it. The Prob(p) is employed to improve the cache efficiency and to minimize caching redundancy [9].

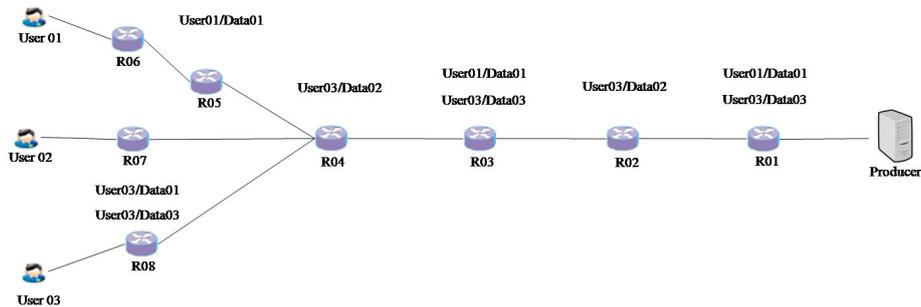


Fig. 3. Illustrative example of Prob(p) caching strategy with  $p=0.5$

Figure 3 presents an illustrative example of the Prob (p) strategy with a  $p$ -value equal to 0.5. We reserved the same scenario as the LCE strategy. User01 requests Data01. It duplicated on the NDN cache routers with value  $p$  less than 0.5: R01, R03, and R05. User03 requests Data01. It retrieved from R03 and duplicated on each NDN cache router with a value  $p$  small than 0.5 on the delivery path. User03 requests Data02, and Data03. They fetched from the Producer and duplicated on each NDN cache router whose generated  $p$ -value is small than to define value  $p$ . Data02 duplicated on routers R02 and R04, Data03 copied on R01, R03, and R08.

The Prob(p) strategy stores diverse data packets at each cache router on the delivery path. It enhances the cache efficiency, converges to save popular content and enhances the cache-hit ratio. However, its performance relies on the defined probability  $p$ -value. The Prob(p) behaves as the LCE policy when the probability  $p$ -value is equal to one [13].

**Probcache** [14] is a probabilistic caching policy. The Probcache strategy computes the content caching probability by multiplying the time-in by the cache weight based on the TSI and the TSB values. The two values obtained as follows: include a Time-Since-Inception (TSI) field in the interest packet header and include a Time-Since-Birth (TSB) field in the data packet header. TSI value set to zero when the user sends an interest packet. At every hop, the TSI value is incremented by one. The content producer sets the TSB value to zero. The data packet returned to the requester across the delivery path. The TSB value is incremented by one at each reached NDN cache router. The Probcache cache strategy aims to store content near to the user to guarantee resource allocation fairness, to reduce content redundancy, and to save popular content. However, the computation requires the knowledge of remaining cache size for each cache router on the delivery path. The computations on each NDN cache router consume the NDN cache router resources and increase the response time. The fixed value of the mean time gives unreal evaluation.

**Probcache+** [15] is an enhanced version of the Probcache. It has a slight difference. The cache weight powered by the TSB value. However, the Procach+ still suffers from the same drawbacks as the Probcache.

**Intra-AS Cache Cooperation** [16] is an intra-domain cache cooperation policy that adds to each NDN cache router two tables: Local Cache Summary Table (LCST) and Exchange Cache Summary Table (ECST). Periodically, the NDN cache routers announce their LCST to their direct neighbors. The policy reduces the cache redundancy. However, the regular exchange of lists wastes the network resources and slows down the network. Some records in the ECST table are obsolete.

**In-Network Caching for Information-Centric Networking with Partitioning and Hash-Routing: CPHR** [17] is a collaborative in-network caching strategy with content space partitioning and hash-routing information-centric networking. The CPHR modifies the NDN cache router FIB structure by adding two new fields (Cache router name, Egress content router) to allow mapping content via the hash function. The CPHR reduces the cache redundancy on the network. However, the hash mechanism demands centralized control. It incurs high overhead.

**Cluster-based in-networking caching for content-centric networking** [18] is a cluster base caching strategy with a virtual distributed hash function to manage the cluster resources. The cluster cache routers utilize the same hash function to compute the cache router location. It improves the cache diversity, enhances the hits ratio, and reduces the cache redundancy. The cluster construction is very interesting. It replaces the Euclidean distance in the k-medoid clustering algorithm with new distance to reflect the real relationship between the NDN cache routers in the network. However, the stretch ratio increases, due to the hash function utilization to manage the stored content at the cluster cache routers. It has limited scalability and no-cache router location consideration. The hash routing schema incurs high link load. The hash function increases the stretch ratio.

**Caching strategy based on the hierarchical cluster for named data networking** [19] is a two-layer hierarchical cluster-based caching strategy. The Core Layer contains the NDN cache routers that focus only on content routing. The Edge Layer contains the NDN cache routers that store contents near to users. It takes into consideration

the cache router placement and cache popularity. It reduces the cache redundancy. However, the cluster heads have numerous computations that slow down the network. In case of failure, the network paralyzed. The frequent exchange messages introduce extra overhead and flood the network. The arbitrary selection of some parameter values and the static update period incurred inefficient performance. The search for stored contents in the cluster increases the response time. Only the number of shortest paths passing by the caching router defines the cache router's importance. It would be interesting to add other parameters for a more relevant evaluation.

The proposed NECS for NDN architecture overcomes the shortcomings and limitations discussed above. The main idea of this work published in a previous article [20]. It based on clustering the network.

In each cluster, three cache routers selected based on pertinent criteria. The first router called the main cache router. The clustering offers the NDN architecture high scalability and efficiency. The cache routers selection allows the NDN architecture to minimize the content redundancy, to increase the cache-hit ratio, to increase the content diversity, to remove redundant traffic and, to optimize network resources.

### **3 The proposed approach**

In this section, we summarize the NECS approach for Named Data Networking architecture. We highlight the introduced modifications, present the enhanced clustering algorithm and briefly describe the cache routers selection strategy, and the cache management within the clusters.

#### **3.1 Clustering**

The network is divided into several clusters using the clustering mechanism presented in previous published works [20], [21].

The network divided into small sets using a modified k-medoid clustering algorithm with relevant parameters, namely, delay, bandwidth, cache size, and number of hops. These parameters give more realistic and efficient clusters construction, not arbitrary construction.

Each cluster contains at least one border node, which considered as an NDN cache router. It has a direct link with an NDN cache router that belongs to another cluster.

#### **3.2 Cache routing selection**

Let us summarize briefly the cached routers selection mechanism which was presented and demonstrated in [20], [21]. The network clustered. In each cluster, we define three cache routers: the main, the second, and the third. This selection based on three criteria: congestion level, number of connections, and distance from the centroid cluster. We utilize adequate Multi-Attribute Decision-Making methods (MADM) that ranks all the cluster cache routers by descending order from the best to the worst, based on the selected criteria.

### 3.3 Cache management in the network

For cluster management, we proceed as follows: when the cluster's user sends an interest packet or the cluster's border router cache receives an interest packet, they send the interest packet to the main router cache via the shortest path.

The main router cache checks its storage. The found requested content returns to the cluster's user. Otherwise, it redirected to a border router cache via the shortest path. The border router cache redirects the interest packet to the next cluster via the cluster's border router cache.

When a data packet arrives at the cluster's border router cache, it redirected to the cluster's main router cache. Then, the cluster's main router cache stores a copy within its cache memory and sends the data packet to the user who requests it.

Inside each cluster, the contents of the secondary router cache and those of the tertiary router cache are using as a backup to replace the primary router cache in the event of a failure. No cache routers except the main cache router store any content.

When the main cache router store is full, the cache replacement policy Least Frequently Used (LFU) is applied.

The above-proposed mechanism allows reducing cache redundancy in-network, eliminating unnecessary traffic, improving response time, increasing contents diversity, optimizing resource use, minimizing bandwidth consumption, and releasing the overload on the servers. It has low complexity with high performance.

## 4 Performance evaluation

In this section, we describe the experimental setup. We present our simulation results and compare the performance of the proposed approach with other content caching strategies, namely, the Leave a Copy Everywhere (LCE), Random, and Prob(p) with two values of the probability parameter  $p=0.1$  and  $p=0.5$  respectively, store size 10% and 50% of the requested contents. Performance compared in cache-hit ratio.

### 4.1 Simulation setup

The proposed caching strategy is implemented and simulated under the NdnSIM simulator [22], [23]. This simulator relies on the Network Simulator NS3 for network simulations. It designed for the NDN architecture simulations. We compared the obtained results of our proposed caching strategy with the results of three in-network caching strategies. The three caching strategies performed using the NdnSIM with the same topology and the same simulation parameters. Table 1 summarizes the main simulation parameters.

The network topology contains 21 NDN cache routers (nodes). The cache memory configured at 200 GB. The network topology contains 08 data requesters and one data provider. The bandwidth configured to 100 Mbps. The requesters send 100 interest packets per second.

**Table 1.** Simulation parameters

<b>The <math>\alpha</math> parameter of the Zipf distribution</b>	$\alpha = \{0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3\}$
<b>The requested contents amount (catalog size)</b>	$10^3, 10^4, 10^5, 10^6, 10^7$
<b>Content store size</b>	200 GB
<b>Bandwidth</b>	100 Mbps
<b>Topology size</b>	21
<b>Consumer and producer</b>	(08,01)
<b>Simulator</b>	NdnSIM

The NECS performance against the performance of the three selected cache strategies (LCE, Prob (0.1), Prob (0.5), and Random). The mechanisms of these strategies explained with illustration examples in section II. The NECS performance analyzed with different requested contents size and different contents popularity (represented by  $\alpha$  parameter of the Zipf distribution).

The Internet content traffic is classified into four types of content categories: Web content, User Generated Content (UGC), File sharing, and Video on Demand VoD [24]. The popularity of Internet content traffic categories is modeled by Zipf distribution [25]. The content popularity demonstrated that it generally follows Zipf distribution [26], [27]. The  $\alpha$  parameter value of the Zipf law is related to the behavior of user requests, where slightly high values indicate that requests are more concentrated on some contents namely, how often each particular content is required. Different scenarios and applications may require different values of  $\alpha$  parameter. In the literature,  $\alpha$  parameter varies largely from 0.5 to 3.5. For instance,  $\alpha$  parameter value varies between 0.8 and 1.2 in [28].  $\alpha$  parameter value ranges from 0.65 to 2.5 in [29]. The Daily motion catalog is determined with  $\alpha$  equal to 0.88 and the PirateBay catalog is determined with  $\alpha$  equal to 0.75 [30].  $\alpha$  parameter value ranges between 0.65 and 1.0 according to Video on Demand china statistic [31].

In simulations, the Zipf probability distribution used as a popularity model with different  $\alpha$  parameter values between 0.5 and 3 to compare extensively the behavior of the NECS proposed caching strategy with the selected caching strategies.

For robust analysis, the present study utilizes a reasonable interval for  $\alpha$  parameter value of Zipf distribution to simulate popular and unpopular Internet contents. The selected  $\alpha$  parameter values are as follows: 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, and 3.

For a relevant evaluation, we study the impact of different requested content amounts on the performance of the NECS proposed approach. We selected different requested content amounts range from a small amount to a large amount. The requested content amounts are as follows:  $10^3$ ,  $10^4$ ,  $10^5$ ,  $10^6$ , and  $10^7$ .

The cache-hit ratio metric is used to evaluate the NECS performances and against the selected cache strategies (LCE, Random, Prob (0.1), and Prob (0.5)).

The cache-hit ratio metric is an essential parameter to measure the efficiency and performances of any caching strategy in the NDN architecture. A high cache-hit ratio means more requests are satisfied from network cache routers. The cache-hit ratio is the most commonly used metric for evaluating network performances.

The cache-hit ratio represents the average number of found content hits to satisfy interest packets from a cache router. The cache-hit ratio expressed as follows:

$$CacheHitRatio = \frac{\sum_{i=1}^n Hit_i}{\sum_{i=1}^n Hit_i + \sum_{i=1}^n Miss_i} \quad (1)$$

Where

- n is the total number of the network cache routers,
- $\sum_{i=1}^n Hit_i$  Contents sum satisfied from the network cache routers,
- $\sum_{i=1}^n Miss_i$  Contents sum dissatisfied from the network cache routers.

#### 4.2 Simulation results

In this subsection, we study the impact of different Zipf parameter  $\alpha$  values on the performances of NECS’s proposed caching strategy against the selected caching strategies. We also study the impact of different requested content amounts on the NECS proposed caching strategy performances against the selected caching strategies.

The performances evaluation of NECS strategy against the selected cache strategies relies on the cache-hit ratio metric.

Figures 4–8 present the simulation performance results of the four caching strategies (NECS, LCE, Prob (0.1), Prob (0.5), and Random) with different requested contents amounts over Zipf parameter  $\alpha$  values ranging from 0.5 to three.

Each Figure represents a graph that reflects the simulation results for the four approaches with the same values of  $\alpha$  parameter for a single requested content amount. Each curve represents a single cache strategy. The horizontal axis represents the different values of  $\alpha$  parameter and the vertical axis, the cache-hit ratio according to a specific requested content amount.

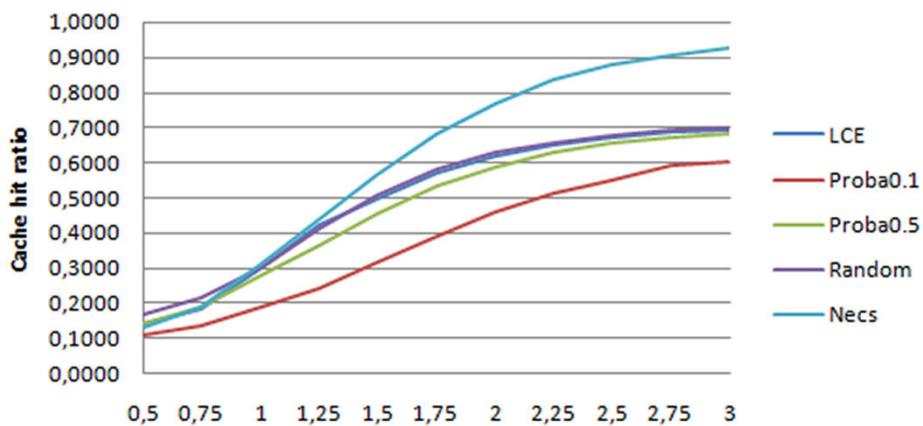


Fig. 4. Requested contents amount  $10^3$

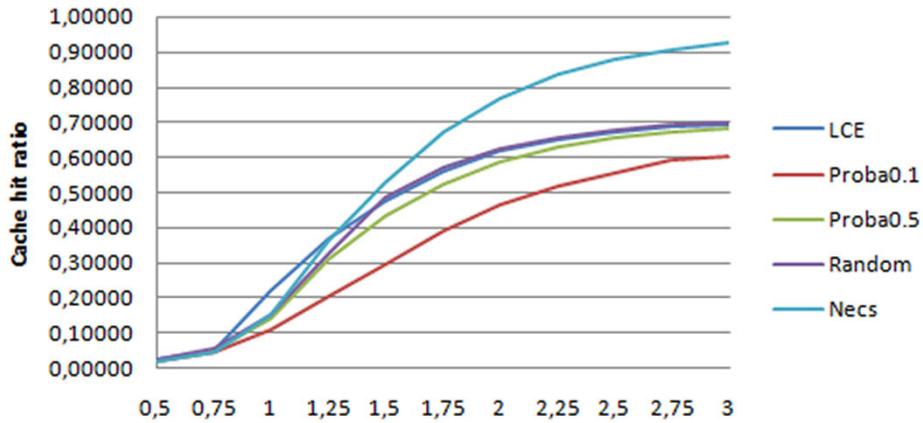


Fig. 5. Requested contents amount  $10^4$

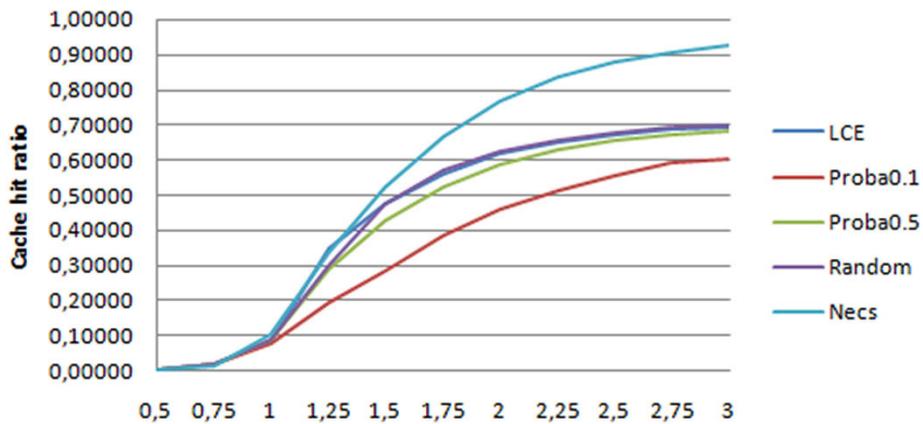


Fig. 6. Requested contents amount  $10^5$

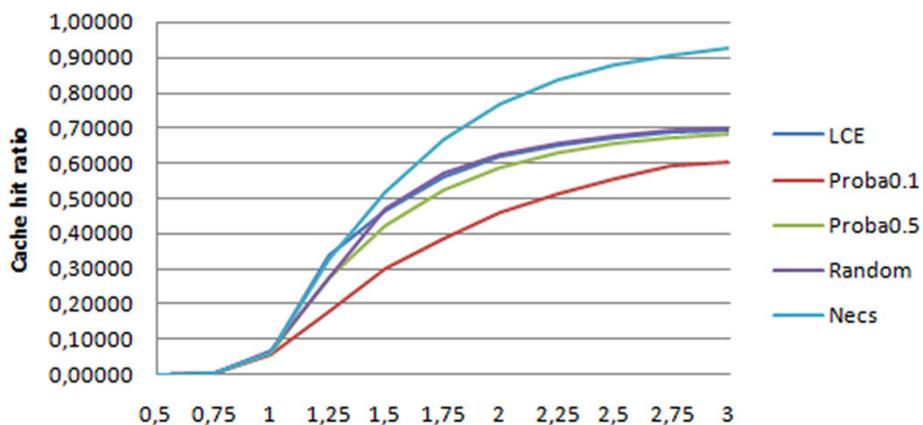


Fig. 7. Requested contents amount  $10^6$

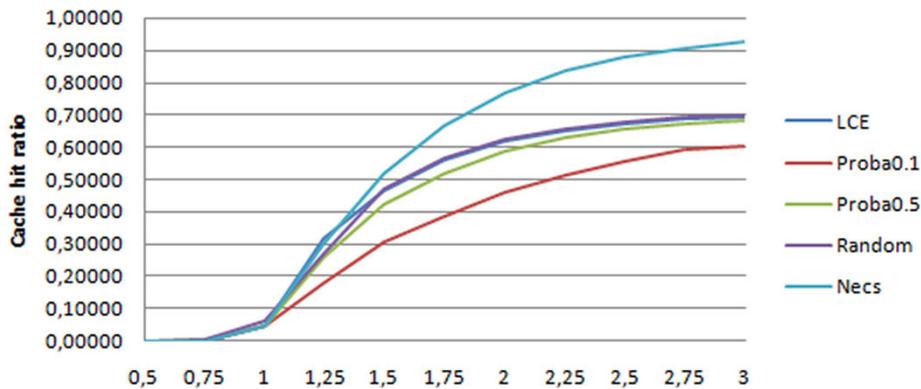


Fig. 8. Requested contents amount  $10^7$

From the results obtained by the conducted simulations, we observe that the requested content amount has no significant impact on the cache-hit ratio as Zipf distribution parameter  $\alpha$ . It presents the content Internet traffic model.

We notice in Figures 4–8 that the cache-hit ratio increases for all the caching strategies as Zipf parameter  $\alpha$  value increases. It explained by the fact that by increasing the Zipf parameter  $\alpha$  value only a small set of contents requested more frequently. It favors retrieving the required contents from NDN cache routers rather than retrieving the required contents from the data provider.

Case requested contents amount and  $\alpha$  parameter

- Requested contents amount  $\leq 10^3$  and  $0.5 \leq \alpha \leq 0.8$ : The Random cache strategy is the best among the four caching approaches because this cache strategy records the highest cache-hit rate compared to other caching strategies. The Prob(0.5) at the second place strategy. The NECS caching strategy and the LCE caching strategy found at the third place. The Prob (0.1) cache strategy found at the last place with the lowest recorded cache-hit values (see Figure 4).
- Requested contents amount  $\leq 10^3$  and  $0.8 < \alpha \leq 3$ : The NECS is the best caching strategy. It records the highest cache-hit ratio values compared to the other caching strategies. The LCE cache strategy and the Random cache strategy rank second. The Prob (0.5) cache strategy rank third. The Prob (0.1) cache strategy ranks last with the lowest cache-hit rate ratio values (see Figure 4).
- Requested contents amount =  $10^4$  and  $0.5 \leq \alpha \leq 1.1$ : The LCE cache strategy is the best caching strategy. It records the highest cache-hit ratio compared to the other selected caching strategies. The NECS strategy rank second. The Random caching strategy ranks third. The Prob (0.5) caching strategy ranks fourth. The Prob (0.1) strategy ranks fifth (see Figure 5).
- Requested contents amount =  $10^4$  and  $1.1 < \alpha \leq 3$ : The NECS strategy is the best compared to the remaining caching strategies. It records the highest cache-hit ratio values. The LCE and the Random caching strategies rank second. The Prob(0.5) cache strategy ranks third. The Prob(0.1) cache strategy ranks last (see Figure 5).

- Requested contents amount =  $10^5$  and  $0.5 \leq \alpha \leq 3$ : The NECS strategy is the best caching strategy. It records the highest cache-hit ratio compared to the remaining selected caching strategies. We notice a clear performance advantage over the other selected caching strategies. The LCE caching strategy ranks second with the Random caching strategy. The Prob (0.5) caching strategy ranks third. The Prob(0.1) caching strategy ranks fourth with the lowest cache-hit ratio values (see Figures 6–8).

The NECS caching strategy is the adequate caching strategy for the Internet. It shows efficient performance at the large scale of requested contents. It is the Internet case.

The NECS adequate for a large requested contents amount with the different values for  $\alpha$  parameter (popular and unpopular contents). The NECS also adequate for small and medium requested contents amount with somewhat high values of  $\alpha$  parameter.

Our simulations show that the default caching strategy of the NDN architecture is not the suitable caching mechanism for the NDN architecture. It does not achieve a high cache-hit rate. Similarly, the Random caching approach and the Prob (p) caching strategy do not achieve high cache-hit rate, in which its performance relies a lot on the p probability parameter as it shown by our simulation results.

The NECS caching approach greatly improves the cache-hit rate for the NDN architecture. The content retrieved from the selected cache routers is according to relevant criteria. The NECS reduces retrieving contents from the content provider.

Our simulation results show the important impact of the cache routers selection to store contents in strategic locations to satisfy future requests. They also show the impact on improving the caching strategy performance and effectiveness and thus, the NDN architecture efficiency. Therefore, the cache router's choice to store copies of the requested data is a critical task.

The NECS caching strategy gives satisfactory and encouraging results. The appropriate selection of cache routers to store the requested content gives the NECS cache strategy high performances. It saves the network bandwidth, optimizes the network resources, eliminates unnecessary network traffic, reduces the cache redundancy, and increases the content diversity.

## 5 Conclusion

In this paper, we present the main concept of the proposed NECS caching strategy based on the clustering mechanism and caching routers selection to store contents. We also present the results obtained by our conducted simulation study for the NECS caching strategy and compare its performances against three selected caching strategies, namely, the Leave a Copy Everywhere (LCE), Random, and Prob(p).

The evaluation results clearly show significant benefits of the NECS caching strategy and its superiority over LCE for large requested contents amount. The LCE strategy is the default caching strategy of the NDN architecture.

We believe that the NECS caching mechanism will play a key role in NDN architecture. It can easily retrieve content from the nearest main cache router, eliminate redundant traffic, improve content diversity, minimize content redundancy, optimize

the network resources utilization, and alleviate the burden on the content provider. The NECS strategy presents a promising caching solution for the NDN architecture.

In future work, we plan to compare the NECS caching strategy to other NDN caching mechanisms with additional metrics and to assess the performance of our strategy under other network topologies.

## 6 References

- [1] “Domo Resource–Data Never Sleeps 8.0.” <https://www.domo.com/learn/data-never-sleeps-8>, released August 11, 2020, (accessed Oct. 31, 2020).
- [2] “Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper,” Cisco. <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>. Updated: March 9, 2020, (accessed Oct. 31, 2020).
- [3] Y. Yamamoto, J. Takemasa, Y. Koizumi, and T. Hasegawa, “Analysis on caching large content for information-centric networking,” in Proceedings of the 5th ACM Conference on Information-Centric Networking, New York, NY, USA, Sep. 2018, pp. 24–30, DOI: <https://doi.org/10.1145/3267955.3267957>
- [4] M. A. Naeem, M. A. U. Rehman, R. Ullah, and B. Kim, “A Comparative Performance Analysis of Popularity-Based Caching Strategies in Named Data Networking,” IEEE Access, vol. 8, pp. 50057–50077, 2020, DOI: <https://doi.org/10.1109/ACCESS.2020.2980385>
- [5] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking Named Content,” in Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, New York, NY, USA, 2009, pp. 1–12, DOI: <https://doi.org/10.1145/1658939.1658941>
- [6] L. Zhang et al., “Named data networking,” ACM SIGCOMM Comput. Commun. Rev., vol. 44, no. 3, p. 8, 2014. <https://doi.org/10.1145/2656877.2656887>
- [7] D. Rossi and G. Rossini, “Caching performance of content-centric networks under multi-path routing (and more),” p. 9, 2011.
- [8] N. Laoutaris, H. Che, and I. Stavrakakis, “The LCD interconnection of LRU caches and its analysis,” Perform. Eval., vol. 63, no. 7, pp. 609–634, Jul. 2006, DOI: <https://doi.org/10.1016/j.peva.2005.05.003>
- [9] H. Jin, D. Xu, C. Zhao, and D. Liang, “Information-centric mobile caching network frameworks and caching optimization: a survey,” EURASIP J. Wirel. Commun. Netw., vol. 2017, no. 1, p. 33, Feb. 2017, DOI: <https://doi.org/10.1186/s13638-017-0806-6>
- [10] C. Bernardini, T. Silverston, and O. Festor, “MPC: Popularity-based caching strategy for content-centric networks,” in 2013 IEEE International Conference on Communications (ICC), Budapest, Hungary, Jun. 2013, pp. 3619–3623, DOI: <https://doi.org/10.1109/ICC.2013.6655114>
- [11] K. Cho, M. Lee, K. Park, T. T. Kwon, Y. Choi, and Sangheon Pack, “WAVE: Popularity-based and collaborative in-network caching for content-oriented networks,” in 2012 Proceedings IEEE INFOCOM Workshops, Mar. 2012, pp. 316–321, DOI: <https://doi.org/10.1109/INFCOMW.2012.6193512>
- [12] M. A. Naeem, S. A. Nor, S. Hassan, and B.-S. Kim, “Performances of Probabilistic Caching Strategies in Content Centric Networking,” IEEE Access, vol. 6, pp. 58807–58825, 2018, DOI: <https://doi.org/10.1109/ACCESS.2018.2872614>
- [13] M. Zhang, H. Luo, and H. Zhang, “A Survey of Caching Mechanisms in Information-Centric Networking,” IEEE Commun. Surv. Tutor., vol. 17, no. 3, pp. 1473–1499, 2015, DOI: <https://doi.org/10.1109/COMST.2015.2420097>

- [14] I. Psaras, W. K. Chai, and G. Pavlou, “Probabilistic in-network caching for information-centric networks,” 2012, p. 55, DOI: <https://doi.org/10.1145/2342488.2342501>
- [15] I. Psaras, W. K. Chai, and G. Pavlou, “In-Network Cache Management and Resource Allocation for Information-Centric Networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 2920–2931, Nov. 2014, DOI: <https://doi.org/10.1109/TPDS.2013.304>
- [16] J. M. Wang, J. Zhang, and B. Bensaou, “Intra-AS cooperative caching for content-centric networks,” 2013, p. 61, DOI: <https://doi.org/10.1145/2491224.2491234>
- [17] S. Wang, J. Bi, J. Wu, and A. V. Vasilakos, “CPHR: In-Network Caching for Information-Centric Networking With Partitioning and Hash-Routing,” *IEEE ACM Trans. Netw.*, vol. 24, no. 5, pp. 2742–2755, Oct. 2016, DOI: <https://doi.org/10.1109/TNET.2015.2480093>
- [18] C. Li and K. Okamura, “Cluster-based In-networking Caching for Content-Centric Networking,” p. 9, 2014. <https://doi.org/10.1109/ICT.2014.6845152>
- [19] H. Yan, D. Gao, W. Su, C. H. Foh, H. Zhang, and A. V. Vasilakos, “Caching Strategy Based on Hierarchical Cluster for Named Data Networking,” *IEEE Access*, vol. 5, pp. 8433–8443, 2017, DOI: <https://doi.org/10.1109/ACCESS.2017.2694045>
- [20] N. E. H. Fethellah, H. Bouziane, and A. Chouarfia, “New Efficient Caching Strategy Based on Clustering in Named Data Networking,” *Int. J. Interact. Mob. Technol. IJIM*, vol. 13, no. 12, pp. 104–119, Dec. 2019. <https://doi.org/10.3991/ijim.v13i12.11403>
- [21] N. E. H. Fethellah, H. Bouziane, and A. Chouarfia, “New efficient Caching strategy based on clustering in Named Data Networking,” in *1st International Conference on Networking Telecommunications, Biomedical Engineering and Applications (ICNTBA’19)*, Boumerdes, Algeria, Nov. 2019, vol. 1, pp. 26–32.
- [22] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, “ndnSIM 2: An updated NDN simulator for NS-3,” p. 8, 2016.
- [23] S. Mastorakis, A. Afanasyev, and L. Zhang, “On the Evolution of ndnSIM: an Open-Source Simulator for NDN Experimentation,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 47, no. 3, pp. 19–33, Sep. 2017, DOI: <https://doi.org/10.1145/3138808.3138812>
- [24] A. Lareida, R. Pernischova, B. B. Rodrigues, and B. Stiller, “Abstracting .torrent content consumption into two-mode graphs and their projection to content networks (ConNet),” in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, May 2017, pp. 151–159, DOI: <https://doi.org/10.23919/INM.2017.7987275>
- [25] G. Hasslinger, K. Ntougias, and F. Hasslinger, “Performance and Precision of Web Caching Simulations Including a Random Generator for Zipf Request Pattern,” in *Measurement, Modelling and Evaluation of Dependable Computer and Communication Systems*, Cham, 2016, pp. 60–76, DOI: [https://doi.org/10.1007/978-3-319-31559-1\\_7](https://doi.org/10.1007/978-3-319-31559-1_7)
- [26] L. Breslau, Pei Cao, Li Fan, G. Phillips, and S. Shenker, “Web caching and Zipf-like distributions: evidence and implications,” in *IEEE INFOCOM ’99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No.99CH36320)*, Mar. 1999, vol. 1, pp. 126–134 vol.1, DOI: <https://doi.org/10.1109/INFCOM.1999.749260>
- [27] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, “I tube, you tube, everybody tubes: analyzing the world’s largest user-generated content video system,” in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, New York, NY, USA, Oct. 2007, pp. 1–14, DOI: <https://doi.org/10.1145/1298306.1298309>
- [28] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, “Impact of traffic mix on caching performance in a content-centric network,” Mar. 2012, pp. 310–315, DOI: <https://doi.org/10.1109/INFCOMW.2012.6193511>
- [29] G. Rossini and D. Rossi, “A dive into the caching performance of Content-Centric Networking,” Sep. 2012, pp. 105–109, DOI: <https://doi.org/10.1109/CAMAD.2012.6335307>

- [30] M. Mangili, F. Martignon, and A. Capone, “A comparative study of Content-Centric and Content-Distribution Networks: Performance and bounds,” Dec. 2013, pp. 1403–1409, DOI: <https://doi.org/10.1109/GLOCOM.2013.6831270>
- [31] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, “Understanding user behavior in large-scale video-on-demand systems,” ACM SIGOPS Oper. Syst. Rev., vol. 40, no. 4, pp. 333–344, Apr. 2006, DOI: <https://doi.org/10.1145/1218063.1217968>

## 7 Authors

**Nour El Houda Fethellah** Currently is PHD computer science student in the USTO-MB University of Oran in Algeria. She graduated from the same university with a Master degree in Computer Science Engineering. She is a member of the research team that focused on networking. Email: [nourelhouda.fethellah@univ-usto.dz](mailto:nourelhouda.fethellah@univ-usto.dz)

**Hafida Bouziane** is an Associate Professor lecturer at Computer Science Department Faculty of Mathematics and Computer Science of USTO-MB University Oran—Algeria. She received the PhD degree in Computer Science from USTO-MB in 2014. She is member of LIM (Computer science and mathematics) Laboratory—University Ibn Khaldoun of Tiaret. Her research interests include intelligent system, networking, bioinformatics, and m-learning. Email: [hafida.bouziane@univ-usto.dz](mailto:hafida.bouziane@univ-usto.dz)

**Abdallah Chouarfia** was born in Algeria, graduated with Engineer Degree in Computer Science from National Institute of Computer Science, Algiers, in 1981. He received the PhD degree in Computer Science from the Paul Sabatier University in Toulouse, France in 1983. He was a Senior Lecturer in the Computer Science Department at Sciences and Technology University, Oran Algeria from 1987–2019. He is currently a full Professor at Science and Technology University, Oran Algeria. His research interests include networking, software engineering, information systems interoperability, web engineering, and pervasive systems. He is member of LIM (Computer science and mathematics) Laboratory—University Ibn Khaldoun of Tiaret. [abdallah.chouarfia@univ-usto.dz](mailto:abdallah.chouarfia@univ-usto.dz)

Article submitted 2020-11-24. Resubmitted 2021-05-26. Final acceptance 2021-05-26. Final version published as submitted by the authors.