

NECS-based Cache Management in the Named Data Networking

N. FETHELLAH, H. BOUZIANE, A. CHOUARFIA

Département Informatique

Université des Sciences et de la Technologie d'Oran - Mohamed Boudiaf

Oran, Algérie

Email :{ nourelhouda.fethellah, hafida.bouziane, abdallah.chouarfia }@univ-usto.dz

Abstract—the Information-Centric Networking ICN architectures proposed to overcome the problems of the actual internet architecture. One of the main straight points of the ICN architectures is in-network caching. The effectiveness of the adopted caching strategy, which manages and decides where to store them, influences the performance of the ICN. However, the major issue that faces the caching strategies in the ICN architectures is the strategic selection of the cache routers to store the data through its delivery path. This will reduce congestion, optimize the distance between the consumers and the required data furthermore improve latency and alleviate the viral load on the servers. In this paper, we propose a new efficient caching strategy for the Named Data Networking architecture NDN named NECS, which is the most promising architecture between all the ICN architectures. The proposed strategy reduces the traffic redundancy, eliminates the useless replication of contents, and improves the replay time for users due to the strategic position of cache routers. Besides, we evaluate the performance of this proposed strategy and compare it with three other NDN caching strategies, using the simulator network environment NdnSIM. Based on the simulations carried out, we obtained interesting and convincing results.

Keywords—ICN, NDN, caching strategy, future internet architecture, in-network caching, cache management policies

I. INTRODUCTION

The explosion of Internet usage in everyday life has created a set of new challenges. Each minute, more volume of information is broadcasted over the Internet People around the world increasingly move 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 size 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 the 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[1] 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 an online meeting. On LinkedIn, 69,444 people apply for jobs. On the Amazon website, 6,659 Amazon packages shipped. On YouTube, 500hrs video viewed.

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 earlier, the massive broadcast on the Internet while the architecture is unresponsive. 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 is a core Internet architecture. Among the adopted techniques, the Information-Centric Networking (ICN) architectures try 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[1].

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

The NDN default caching mechanism consists of 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. The NDN architecture would be more efficient and robust by suggesting an appropriate content caching mechanism, which takes into consideration to determine the important cache routers to allow them to store the disseminated content across the network.

In previously published work, we investigated these requirements and the trade-off for making this decision and we present the main idea of a suggested caching strategy. The proposed approach contains two main parts, which are clustering the network and caching routers selection. The improved K-medoid algorithm does the clustering. In each

resulting cluster, the three most important cache routers are selected based on three pertinent criteria using relevant Multi-Attribute Decision-Making (MADM) methods. The three criteria are the distance between a router and its cluster centroid (dis), the number of neighbors (nbrn), and congestion level (cl)).

In the present paper, we prove the efficiency of the proposed caching strategy for the NDN architecture, which we give as a name NECS, which means New Efficient Caching Strategy. The contributions of this paper are presented as follows:

- Presenting the cache management process of the NECS caching strategy.
- 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 a higher cache-hit ratio and shorter potential stretches (hops). It is a well-suited alternative to the actual default caching strategy of the NDN architecture since it improves network performance and resilience.

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.

II. 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.

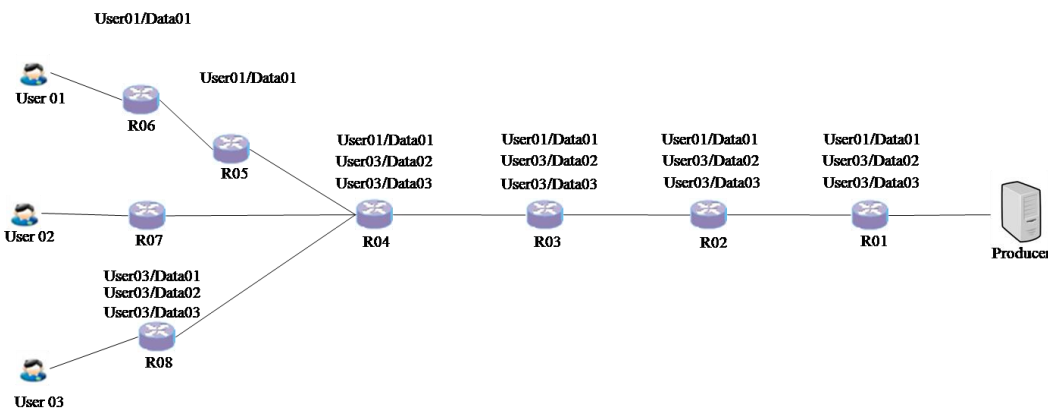


Fig. 1 Illustrative example of LCE caching strategy

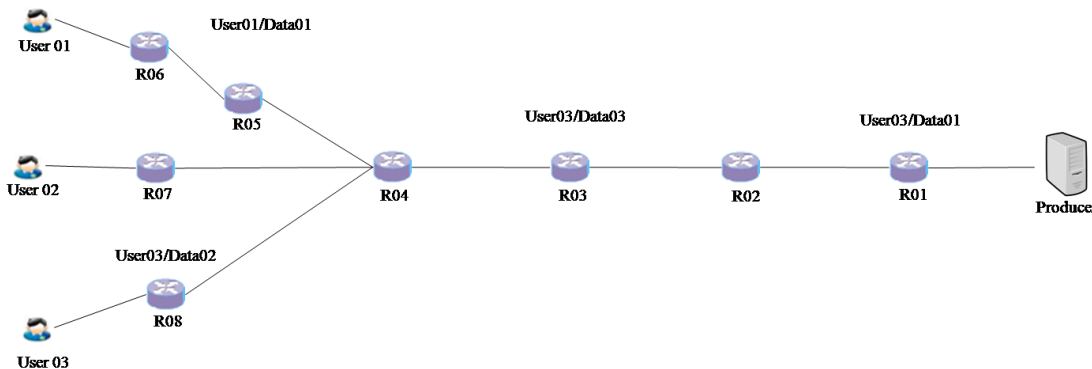


Fig. 2 Illustrative example of Radom caching strategy

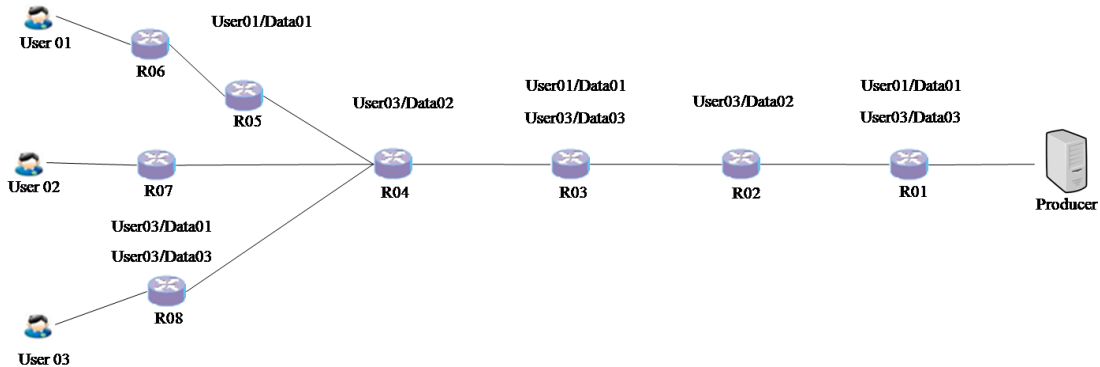


Fig. 3 Illustrative example of Prob(p) caching strategy with p=0.5

To get the benefits of NDN architecture, efficient cache management is required. 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[2], [3] 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 presents an illustrative example of the LCE strategy. There are three users User01, User02, User03, and one Producer. The User01 requests the Data01. This data replicated on all cache routers located between the Producer and the User01 (R01, R02, R03, R04, R05, R06). User02 does not request any data. The User03 requests Data01, Data02, and Data03 respectively. Data01 fetched from the cache router R04 and duplicated on all the cache routers located along the delivery path. Data02 and Data03 are obtained from the Producer and duplicated on the same delivery path. When the User03 requests once more Data01, Data02, or Data03, it is satisfied from 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 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[4] is a cache management policy. The LCD strategy consists of 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 a long time to place content near to the user.

Move a Copy Down MCD[5] 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 caching 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[6]. The repetitive requests for the same content consume the network resources.

Most Popular Content: MPC[7] 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 size in the popularity table (PT). Requested content is considered popular when its request number is equal to or greater than 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 a 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[8]. 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 is randomly selected. Fig. 2 presents an illustrative example of the Random strategy. We take the same scenario as the LCE strategy. The User01 requests the Data01. The Producer satisfies it, a copy of the Data01 is stored on the router R05. The User03 requests the Data01. The NDN cache router R05 does not serve the request. It is not located on the same common part of the delivery path. Data01 is retrieved from the Producer and a copy is stored on the router R01. The User03 requests the Data02. It is satisfied by the Producer and a copy is stored on the router R03. The User03 requests the Data03. It is fetched from the Producer and a copy is saved on the router R08. The Random strategy is an autonomous randomly cache content policy. It reduces content duplication and increases content diversity. It achieves a 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[9].

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 small than the p -value then 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[6]. Fig. 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. The User01 requests the Data01. This one is duplicated on the routers with a p -value less than 0.5. Data01 is stored on the routers R01, R03, and R05. The User03 requests the Data01. It is retrieved from the router R03 and duplicated on each router with a p -value small than 0.5 on the delivery path. The User03 requests the Data02, then the Data03. They are fetched from the Producer and duplicated on each router whose generated p -value is less than the defined p -value. Respectively, the Data02 on the routers R02 and R04, and Data03 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[10].

Probcache[11] 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. It includes a Time-Since-Inception (TSI) field in the interest packet header and a Time-Since-Birth (TSB) field in the data packet header. The TSI value is 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 is sent to the data requester across the delivery path. The TSB value is also incremented by one for each NDN cache router reached. The Probcache strategy aims to store content near to the user to guarantee resource allocation fairness, reduce content redundancy, and save popular content. However, the

computation requires the knowledge of the 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 meantime gives unreal evaluation. Probcache+[12] is an enhanced version of the Probcache. It has a slight difference. The cache weight increases by the TSB value. However, the Procach+ still suffers from the same drawbacks as the Probcache.

Intra-AS Cache Cooperation[13] 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 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[14] 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[15] 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 cache diversity, enhances the hits ratio, and reduces cache redundancy. The cluster construction is very interesting. It replaces the Euclidean distance in the k-medoid clustering algorithm with a 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 a high link load. The hash function increases the stretch ratio.

Caching strategy based on the hierarchical cluster for named data networking[16] 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 is 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.

To overcome the shortcomings discussed above, we propose NECS as a new efficient caching strategy for NDN architecture. The main idea of this work based on clustering was published in a previous article [17]. Pertinent criteria selected three cache routers (the main, the first, and the second) for each cluster. The clustering offers the NDN architecture high scalability and efficiency. The cache routers selection allows the NDN architecture to minimize the content redundancy, increase the cache-hit ratio, increase the content diversity, remove redundant traffic and, optimize network resources.

III. THE PROPOSED APPROACH

In this section, we present only the main idea of the NECS caching strategy because we are limited with the allowed number of pages. However, we explain in detail the suggested approach in two previously published works [17], [18]. The NECS caching strategy contains two main parts, which are clustering the network and selecting the three most important cache routers in each resulting cluster.

Clustering

The network is divided into several clusters using the clustering mechanism, which is the improved k-medoid clustering algorithm [15] with appropriate parameters: delay, bandwidth, cache size, and the number of hops. These parameters give more realistic and efficient cluster construction. This clustering algorithm has input the graph of the network and the number of the desired clusters. It has as output the resulting clusters. Each resulting cluster contains at least one border node considered as an NDN cache router. It has a direct link with an NDN cache router that belongs to another cluster. Its basic process is as follows:

- deleting all the routers with only one link,
- Set the K routers with the most number of links as initial centroids of K clusters,
- For each router, compute its distance to each centroid of the K clusters,
- Assign the router into the cluster with the smallest distance value,
- For every single cluster, compute the distance between every two routers.
- Set the router with the smallest distance value as the new centroids of its cluster,
- Repeat all the steps except the two first steps until the centroid is unchanged.

Cache routers selection

We present the main idea of the NECS for the cached routers selection. However, we explain in detail the caching router selection mechanism in two previously published works [17], [18]. In each resulting cluster, the three most efficient cache routers are selected based on three pertinent criteria: congestion level, number of connections, and distance to the centroid cluster.

We utilize two Multi-Attribute Decision-Making methods (MADM), which are the Technique for Order of Preference by Similarity to Ideal Solution method (TOPSIS) combined with the Analytic Hierarchy Process (AHP). The

TOPSIS method allows ranking all the cluster cache routers by descending order from the best to the worst, based on the selected criteria. The AHP method is utilized to evaluate the assigned weights of the selected criteria in the TOPSIS method. For assuring the optimal caching routers selection with the three chosen criteria, we utilize two MADM approaches. The two MADM approaches are the Technique for Order of Preference by Similarity to Ideal Solution method (TOPSIS) combined with the Analytic Hierarchy Process (AHP). The criteria become the distance between a node and its cluster centroid (*dis*), the number of neighbors (*nbrn*), and the congestion level (*cl*). Fig.4. illustrates the process of the cache routers selection.

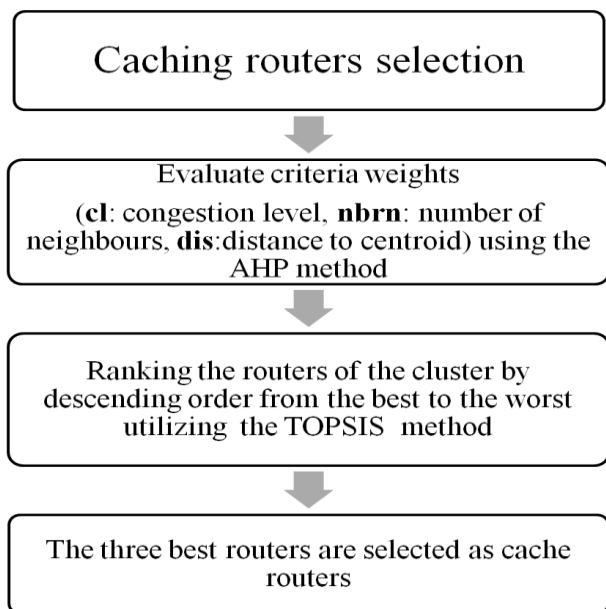


Fig.4 The process of cache router selection

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; if the requested content is found then it returned it to the cluster's user else it is redirected to a border router cache via the shortest path. The border router cache redirects the interest packet to the next cluster via its cluster's border router cache.

When a data packet reaches the cluster's border router cache, it is redirected to the cluster's main router, which stores a copy and sends the packet to the requester user. If the data packet arrives in the main cache router with a full cache, the policy Least Frequently Used (LFU) is applied rather than Least Recently Used policy (LRU), which is the default replacement policy of the NDN architecture. The main cache router of the cluster is the only router that has permission to store incoming contents into the cluster. 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.

The above-proposed mechanism reduces cache redundancy in-network, eliminates unnecessary traffic, improves response time, increases contents diversity,

optimizes resource use, minimizes bandwidth consumption, and releases overload on the servers. It has low complexity with high performance.

IV. PERFORMANCE EVALUATION

In this section, we describe the experimental setup. We present 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. The performance is rated in terms of cache-hit ratio.

Simulation setup

The proposed caching strategy is implemented and simulated under the NdnSIM simulator[19], [20]. The NdnSIM relies on the Network Simulator NS3. We compared the obtained results of the proposed caching strategy to the results of three in-network caching strategies performed using the same simulator, the same topology, and the same parameters. Table I shows the simulation parameters.

TABLE I. SIMULATION PARAMETERS

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

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

The performances of LCE, Prob(0.1), Prob(0.5), and Random strategies compared to the NECS performances. NECS performance analysis takes on different content sizes and popularities (represented by α 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 [21]. The popularity of Internet content traffic categories is modeled by Zipf distribution[22]. The content popularity demonstrated that it generally follows Zipf distribution[23], [24]. The α 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 the α parameter. In the literature, α parameter Zipf varies largely from 0.5 to 3.5. For instance, the α parameter value varies between 0.8 and 1.2 in [25]. The α parameter value ranges

from 0.65 to 2.5 in [26]. The Daily motion catalog is determined with α equal to 0.88 and the PirateBay catalog is determined with α equal to 0.75[27]. The α parameter value ranges between 0.65 and 1.0 according to Video on Demand china statistic[28].

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

For robust analysis, the present study utilizes a reasonable interval for the α parameter value of the Zip distribution to simulate popular and unpopular Internet contents. The selected α 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 sizes on the performance of the NECS proposed approach. We selected different requested content sizes to range from a small size to a large size. 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 performance 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 caching router. (1) expressed the cache-hit ratio.

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

Where

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

Simulation Results

In this subsection, we study the impact of different α Zipf parameter values on the performances of NECS's proposed caching strategy against the selected caching strategies. We also study the impact of different requested content sizes on the NECS proposed caching strategy performances against the selected caching strategies.

The performance evaluation of the NECS strategy against the selected cache strategies relies on the cache-hit ratiometric.

Fig. 5, Fig. 6, Fig. 7, Fig. 8, and Fig. 9 present the simulation performance results of the four caching strategies (NECS, LCE, Prob(0.1), Prob(0.5), and Random) with different requested cache sizes over the α Zipf parameter values ranging from 0.5 to 3.

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

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 α Zipf distribution parameter. It presents the content Internet traffic model.

We notice in Fig. 5, Fig. 6, Fig. 7, Fig. 8, and Fig. 9 that the cache-hit ratio increases for all the caching strategies as α Zipf parameter value increases. The increased α Zipf parameter value means that only a small set of contents is more frequently requested. It favors retrieving the required contents from NDN cache routers rather than retrieving the required contents from the data provider.

Case requested contents size and α parameter

- Requested contents size $\leq 10^3$ 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 are found in third place. The Prob(0.1) cache strategy was found at the last place with the lowest recorded cache-hit values (see Fig. 5).
- Requested contents size $\leq 10^3$ 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 Fig. 5).
- Requested contents size = 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 Fig. 6).
- Requested contents size = 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 Fig. 6).
- Requested contents size = 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 Fig. 7, Fig. 8, and Fig. 9).

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.

When the α Zipf parameter value increases more popular contents are stored at strategic main cache routers and records a high cache-hit ratio.

The NECS adequate for a large requested content size with the different values for α parameter (popular and unpopular contents). The NECS is also adequate for small and medium requested contents size with somewhat high values of α parameter.

Simulations show that the default caching strategy of the NDN architecture is not a suitable caching mechanism for the NDN architecture. It does not achieve a high cache-hit rate. Likewise, the Random caching approach and the Prob(p) caching strategy does not achieve a high cache-hit rate. Their performance relies a lot on the p probability parameter as shown by the results above.

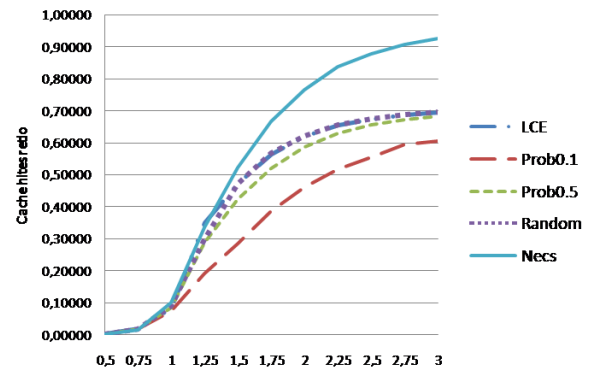


Fig. 7 Required contents size 10^5

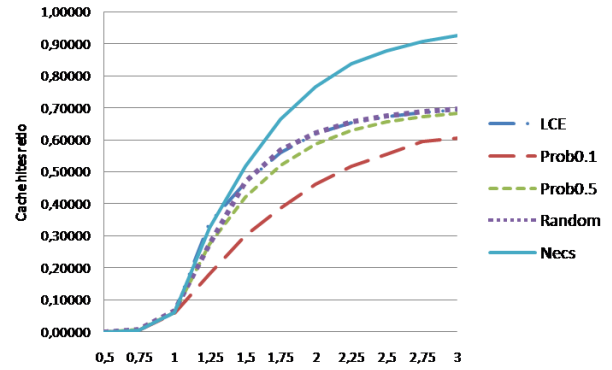


Fig. 8 Required contents size 10^6

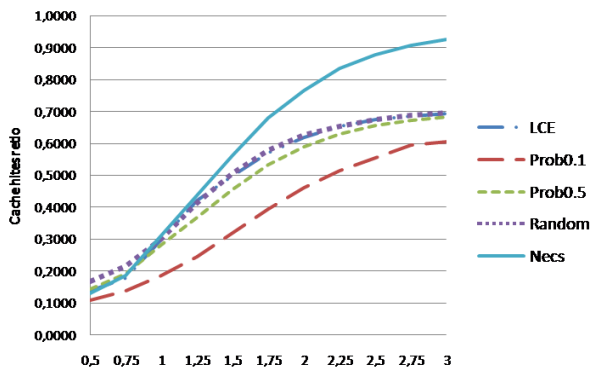


Fig. 5 Required contents size 10^3

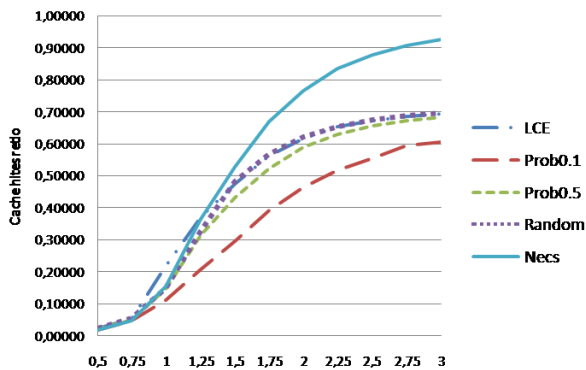


Fig. 6 Required contents size 10^4

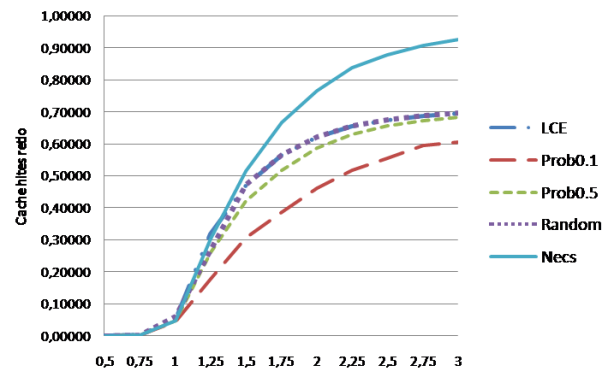


Fig. 9 Required contents size 10^7

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

Simulation results show the important impact of the cache router's selection to store contents in strategic locations to satisfy future requests. They also show the impact on improving the caching strategy performance and effectiveness thus, the NDN architecture efficiency. 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 performance. It saves the network bandwidth, optimizes the network resources, eliminates unnecessary network traffic, reduces cache redundancy, and increases content diversity.

V. CONCLUSION

In this paper, we present the main concept NECS caching strategy. It is based on the clustering mechanism and caching router's selection to store contents. We also present the results obtained by the 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 based on cache-hit ratio and its superiority over LCE for large requested contents size. 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.

The cache-ratio implies, at least, an improvement in the delivery time. In future work, we plan to choose as metric an objective function (cache-hit ratio, delay, and congestion), compare the NECS caching strategy to other NDN caching mechanisms, and assess the performance under other network topologies.

REFERENCES

- [1] 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: 10.1145/3267955.3267957.
- [2] 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: 10.1145/1658939.1658941.
- [3] L. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, p. 8, 2014.
- [4] D. Rossi and G. Rossini, "Caching performance of content-centric networks under multi-path routing (and more)," p. 9, 2011.
- [5] 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: 10.1016/j.peva.2005.05.003.
- [6] 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: 10.1186/s13638-017-0806-6.
- [7] 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: 10.1109/ICC.2013.6655114.
- [8] 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: 10.1109/INFCOMW.2012.6193512.
- [9] "Performances of Probabilistic Caching Strategies in Content-Centric Networking - IEEE Journals & Magazine." <https://ieeexplore.ieee.org/document/8478220> (accessed Dec. 20, 2019).
- [10] 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: 10.1109/COMST.2015.2420097.
- [11] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," 2012, p. 55, DOI: 10.1145/2342488.2342501.
- [12] 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: 10.1109/TPDS.2013.304.
- [13] J. M. Wang, J. Zhang, and B. Bensaou, "Intra-AS cooperative caching for content-centric networks," 2013, p. 61, DOI: 10.1145/2491224.2491234.
- [14] 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: 10.1109/TNET.2015.2480093.
- [15] C. Li and K. Okamura, "Cluster-based In-networking Caching for Content-Centric Networking," p. 9, 2014.
- [16] 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: 10.1109/ACCESS.2017.2694045.
- [17] 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.
- [18] 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, p. P. 26-32.
- [19] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM 2: An updated NDN simulator for NS-3," p. 8, 2016.
- [20] 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: 10.1145/3138808.3138812.
- [21] 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: 10.23919/INM.2017.7987275.
- [22] 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: 10.1007/978-3-319-31559-1_7.
- [23] 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: 10.1109/INFCOM.1999.749260.
- [24] 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: 10.1145/1298306.1298309.
- [25] 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: 10.1109/INFCOMW.2012.6193511.
- [26] G. Rossini and D. Rossi, "A dive into the caching performance of Content-Centric Networking," Sep. 2012, pp. 105–109, DOI: 10.1109/CAMAD.2012.6335307.
- [27] 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: 10.1109/GLOCOM.2013.6831270.
- [28] 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: 10.1145/1218063.1217968.

The Second International Conference on Computer Science's Complex Systems and their Applications ICCSA 2021

Oum El Bouaghi, Algeria
May 25-26th, 2021



ISBN 978-9931-9788-0-0