Article

Controlling the Trade-Off between Resource Efficiency and User Satisfaction in NDNs Based on Naïve Bayes Data Classification and Lagrange Method

Abdelkader Tayeb Herouala [1], Chaker Abdelaziz Kerrache [1], Benameur Ziani [1], Carlos T. Calafate [2],*, Nasreddine Lagraa [1] and Abdou el Karim Tahari [1]

[1] Laboratoire d'Informatique et de Mathématiques, Université Amar Telidji de Laghouat, Laghouat 03000, Algeria; aek.herouala@lagh-univ.dz (A.T.H.); kr.abdelaziz@gmail.com (C.A.K.); bziani@lagh-univ.dz (B.Z.); n.lagraa@lagh-univ.dz (N.L.); k_tahari@yahoo.fr (A.e.K.T.)

[2] Computer Engineering Department (DISCA), Universitat Politècnica de València, 46022 Valencia, Spain

* Correspondence: calafate@disca.upv.es

Abstract: This paper addresses the fundamental problem of the trade-off between resource efficiency and user satisfaction in the limited environments of Named Data Networks (NDNs). The proposed strategy is named RADC (Resource Allocation based Data Classification), which aims at managing such trade-off by controlling the system's fairness index. To this end, a machine learning technique based on Multinomial Naïve Bayes is used to classify the received contents. Then, an adaptive resource allocation strategy based on the Lagrange utility function is proposed. To cache the received content, an adequate content placement and a replacement mechanism are enforced. Simulation at the system level shows that this strategy could be a powerful tool for administrators to manage the trade-off between efficiency and user satisfaction.

Keywords: named data network; cache strategy; machine learning; placement strategy; name classification; resource allocation

1. Introduction

In recent years, new network architectures under the name of Information-Centric Networking (ICN) have attracted the attention of many academics and enterprises. ICN proposes new structures and concepts that can solve the traffic congestion problems caused by massive content distribution in the network [1]. The content is considered a very important factor in ICN, which is the basis for content demand and reception. Among the ICN architectures, NDN is a promising active research project that aims to develop a candidate architecture for the future Internet [2]. NDN follows the same design as the IP system, the difference being that the names of the data replace the IP address. NDN data names are similar to the URL names structure, which better matches the current shift in intranet usage from site-specific to content-specific searches. Routing and data transmission also depend on the content names by implementing the longest prefix corresponding to the requested name prefix [3]. In-network caching is also an important key in NDN, where each router has a cache, and the received data are cached according to a specific caching strategy. The caching mechanism enables future requests to be satisfied by the nearest available routers instead of reaching the end producer of the data [4].

It is obvious that the in-network caching saves limited bandwidth resources and improves network performance; however, in practice, the size of the caches remains limited due to the huge amount of data circulating in the network; therefore, the main challenge is how to use these limited caching resources for massive content while ensuring Quality of Service (QoS). In addition, the administration and allocation of caching resources in the network is also an important issue. These problems have been the subject of several research papers in terms of caching strategies and algorithms; however, many of the

proposed strategies focus on the large network environment and simply consider user satisfaction as a basic factor for caching important data. In practice, these methods are not always effective when dealing with a small limited network, such as a university network, a company network, etc. This type of network usually has a limited set of nodes, and its own data representing its direction, which is often stored in its own servers. In fact, the huge demand for content from social networks and video-on-demand (VoD) has become a large part of the data consumed in everyday life [5,6]. This poses a problem for this type of network, since the owners of these networks will not be able to take advantage of the cached data using existing caching strategies. As an example, the analysis of data traffic at the University Amar Telidji Laghouat shows that the consumed data belonging to the university represents only 4% of the total resource consumption. In this case, with the existing strategies in the NDN, the data related to this network has no advantage in terms of delay, satisfaction, and load on the servers. This looks to be contrary to the primary goals of NDN; therefore, controlling the trade-off between maximum user satisfaction and resource efficiency is one of the most fundamental issues in a limited NDN environment. From the network administrator's perspective, it is very essential to use the network caches efficiently, and the benefits must be maximized. Similarly, from the users' perspective, it is more important that their Quality of Service (QoS) requirements are met adequately. Hence, the question is how to manage this trade-off?

In this paper, we use a proven Bayesian classification-based machine learning method to classify the received data in such a network. Then, using the Lagrange method, we propose a utility function based on a preference parameter that allocates different partitions to the content classes in the routers. A new placement and replacement strategy is also defined for processing the data received by the routers. Finally an exhaustive simulation of the results is presented to demonstrate the performance of the proposed model.

This paper is organized as follows. In Section 2, we review some NDN background. In Section 3, the related work on the NDN caching solution is presented. In Section 4, the proposed solution for resource allocation is described. Simulation results are then presented and discussed in Section 5. Finally, we conclude this study and discuss future research in Section 6.

2. NDN Background

In an NDN communication process, as illustrated in Figure 1, there are two types of packets called interest and data packet. On the one hand, the interest packet mainly contains the name of the content; on the other hand, the data packet mainly contains the name of the content, the encrypted information, and the data. As Figure 1 indicates, NDN is based on three essential modules for processing the requested and received packets. These modules are called Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB). The CS has a role of caching the data packets with some caching strategies in order to satisfy the next requests for the same data. PIT is used to record the interfaces of the incoming interest packets until the interest packet is satisfied, or the recording time of the same packet is expired. The PIT also has a content name aggregation function, in which the interest packets for the same content received from different interfaces will be registered as a single request to avoid the retransmission of similar requests for the same data. The role of the FIB module is to transmit the packets of interest to the producer(s) of the data using packet transmission mechanisms, which are mainly based on the names of the contents.
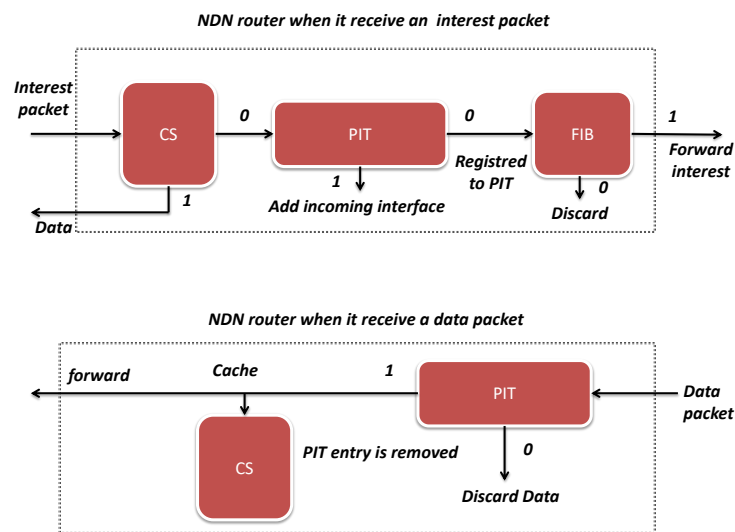
Figure 1. NDN background.

3. Related Work

Based on previous studies on resource allocation in ICNs, we can see a strong similarity between these strategies and content placement strategies. Arguably, the first and best known resource allocation strategy is Leave Copy Every Were (LCE) [3], where resources are shared and cached in all nodes in the return path. There are also other strategies proposed in this area, which can be divided into two categories. Strategies that allocate resources over the entire network or path, and strategies that allocate resources over the node itself.

The first category refers to caching strategies that use coordination between nodes to allocate resources fairly, or to allocate popular resources to nodes that are considered important in the network. In [7], the authors propose a strategy called ProbCache, which approximates the capacity of paths to cache content using the hop count. In this strategy, nodes connected near the content source will be given priority to cache content over other nodes along the path. In [8], the authors propose a strategy that combines routing and resource allocation in the NDN network. This strategy uses a Q-learning algorithm based on the basic components of a semi-Markov decision process called SMDP to prove the best routing and resource allocation path. In [9], the authors propose a joint routing strategy with a resource allocation algorithm for wireless environments, which they called dynamic routing and resource allocation. In [10], the authors propose a resource allocation strategy for a limited network environment. They formulate an optimization problem that attempts to maximize the caching of popular content to reduce latency. At the same time, they minimize the hit variance to achieve a balanced resource allocation among the network nodes. In [11], the authors attempt to divide the network topology into different tiers based on the demands from different interfaces and video applications. The goal is to minimize redundancy and cache popular content closer to the consumers. In [12], the authors studied the object allocation problem using game theory as a distributed many-to-one in ISP networks to improve cache utilization in NDN. In [13], the authors measure the node utilization ratio (NUR), which considers the consumers distribution, the server distribution, and the content routing paths. Then they take the proportion between the NUR and the cache size of the node, along with the transmission ratio of the node, to allocate more contents in the relevant nodes.

Despite the importance of these strategies, they do not offer much to the limited network community, as most of these strategies only rely on the popularity of the data in the network distribution. Further, they do not provide an effective solution for controlling the data that should be cached.

Regarding the second category of proposed strategies, we find that researchers have been more interested in classifying the data to give each class a chance to obtain space to cache its data. The main reason is that sharing data in caches by prioritizing caching for the most popular data prevents the rest of the data from having a chance to be cached. In [14], the authors propose a method for dynamic allocation of cache space. They divide the content into five categories based on the packet characteristics, where each category has a different priority to decide how many items should be cached in each category. In [15], the authors propose a caching strategy called Cache Based on Popularity and Class (CAPIC). This strategy caches a copy of the data one hop closer to the consumer for each request. In addition, when the CS of a node is full, the router checks the class of the received data and replaces it in the portion of the same class. Each class has a portion that changes periodically depending on the calculated popularity of the classes. In [16], the authors divide the received data into two categories of popular and non-popular data based on a calculated threshold where the non-popular data are placed near the consumers, and the popular data are placed at the shortest path that contains a maximum of users that request the same data.

We believe that this type of strategy is better than the first type for limited network environments. The main reason is that, in these strategies, many types of data can be cached, especially the less requested data. Yet, these strategies remain limited because they do not provide a robust classification model for the data, as they classify the data only using their frequencies, and do not focus on the content. Furthermore, the proposed strategies do not provide a control factor that matches the demands of consumers and the needs of the managers of these networks.

4. RADC: Resource Allocation Based Data Classification

4.1. System Model

The system model is much more focused on limited network environments, such as academic and industrial networks, etc. In these types of networks, each area consists of a network administrator, and a set of nodes and server(s). For instance, Figure 2 illustrates a network with four areas denoted $Area = \{Area_i | i \in [1.4]\}$. Routers in the same network area sent periodically the sum of each consumed class to the administrator side. Similarly, the network administrator will calculate the allocated partitions for each class according to the utility function given in Section 4.3. The communication between administrator/routers of the same area can be performed using the IP layer in the NDN hourglass [3]. To classify the received contents, each router in the network will use the proposed classification model (Section 4.2) to assign the contents to the corresponding classes. The nodes outside those networks use the default caching mechanism used in NDN.

4.2. Content Classification

As mentioned above, most of the proposed methods in the NDN do not provide a robust model for content classification, as they typically use simple parameters to classify their content (e.g., request frequency). One of the most important keys introduced by NDN that can be exploited to classify contents efficiently is the data naming, which adopts the same structure as URL naming. The names in NDN are significant sources of information and knowledge; however, processing names is a difficult task regarding their unstructured format. We use a Naïve Bayes (NB) classifier [17], which has been widely used to analyze and solve many scientific problems due to its simplicity, efficiency, and effectiveness [18,19]. Naive Bayes is computationally inexpensive and also needs a very low amount of memory [20]. The role of the NB in this study is to classify the received content into two classes, where the first class represents the consumed data that belong to the server or domain of the addressed network, and the second class is the consumed data that do not bring any benefit to the addressed network. Before starting to use the BN model, a step concerning the preparation of the dataset for the training and testing of the model should be performed. The dataset used should contain a sufficient number of names of the requested contents in each selected class. Once the preparation of the dataset

is complete, the conversion of the dataset into a word count matrix is performed, with each row representing a data name, and the columns representing the features that represent the entire vocabulary existing in the dataset. The model uses the n-gram model [21] to extract more features, and also uses the Term-Frequency Inverse Document Frequency (TF-IDF) model [22] to weigh the selected features.
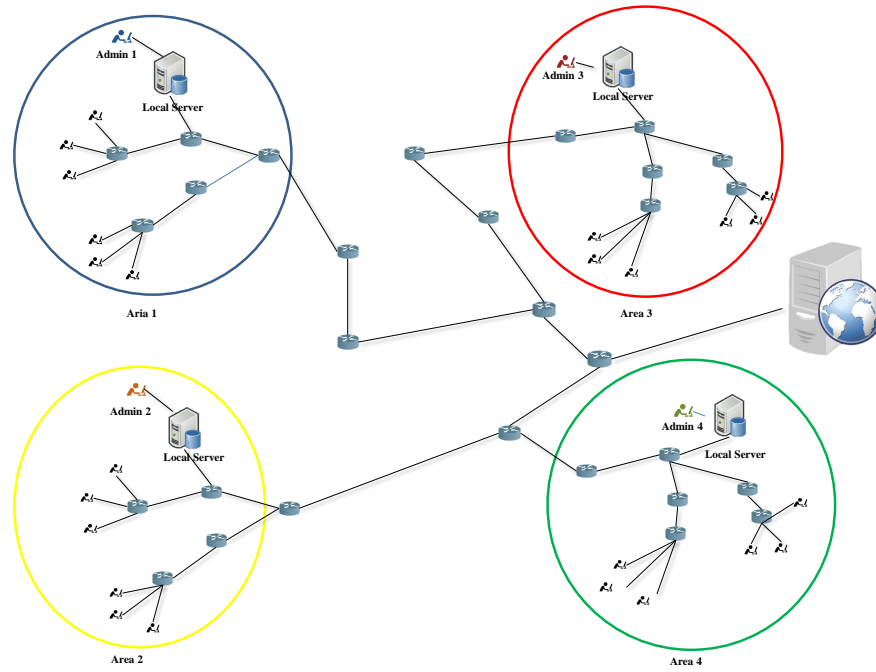

Figure 2. Example of targeted networks.

4.2.1. The High-Level Description of the Naïve Bayes Classifier

If the number of content names (L) fit into s classes where $s \in \{c1, c2, \ldots, cs\}$, the predicted output class is $c \in C$. The Naïve Bayes algorithm can be described as follows:

$$P(c \mid l) = \frac{P(l \mid c)P(c)}{P(l)} \tag{1}$$

where l is the content name, and c indicates classes.

$$C_{\text{Predicted}} = \arg\max_{c \in C} P(l \mid c)P(c) \tag{2}$$

4.2.2. Multinomial Naïve Bayes Classifier

Naive Bayes classifiers differ mainly in the assumptions they make about the probability distribution of $P(l \mid c)$. in text classification such as content names of NDN the multinomial distribution performs well compared to other distributions such as the Gaussian distribution or the Bernoulli distribution [17]. The Multinomial Naïve Bayes algorithm can be written as:

$$P(c \mid l) = \frac{P(c) \prod_{w \in l} P(l \mid c)^{n_{\text{wl}}}}{P(d)} \tag{3}$$

where n_{wl} denotes the number of times that word w occurs in the name, and $P(w \mid c)$ is the probability of observing word w given class c.

$P(w \mid c)$ is calculated as:

$$P(c \mid w_i) = \frac{\text{count}(w_i, c) + 1}{\text{count}(w, c) + V} \tag{4}$$

where $\text{count}(w_i, c)$ is the number of times the word appears in class c in the training dataset, $\text{count}(w, c)$ is the total number of all words appearing for class c, and V is a constant value that represents the number of features (words) that represent the model.

4.3. Resource Allocation

One of these solutions, which is very simple, efficient, and suitable for solving problems of allocation of limited resources between a set of independent activities, is that of Lagrange multipliers. The use of this technique does not guarantee that a solution will necessarily be found for all problems, but it is safe in the sense that any solution found by using them is a real solution [18]. To this end, the aim of this work is to optimize the resource allocation of the two selected classes of data by giving each of them an associated space taking into account two factors. The first one is the priority level of the space associated with data that is not important to the targeted network compared to the second space. The second factor is the frequency of requests received in each class. This is performed within the constraint of the maximum cache space in each router as formulated by the equation below:

$$P \mid \begin{array}{ll} \max_{\tilde{x}} & f(\tilde{x}) \\ \text{u.c} & g(\tilde{x}) = c \end{array} \tag{5}$$

Hence, our problem can be formulated as:

$$P \mid \begin{array}{ll} \max_{C_1, C_2} & f(C_1, C_2) \\ \text{u.c} & g(C_1, C_2) = \text{MaxC} \end{array} \tag{6}$$

where C_1 and C_2 represent, respectively, the space associated to the data that belong to the domain of the targeted network, and the second to the other data; $MaxC$ represents the maximum cache space in the router.

The chosen objective function $f(C_1, C_2)$ and the constrained $g(C_1, C_2)$ that can give the best allocation to the two spaces are formulated according to:

$$P \mid \begin{array}{l} \max_{C_1, C_2} -\alpha C_1 - \beta C_2^k \\ \text{u.c } C_1 + C_2 = \text{MaxC} \end{array} \tag{7}$$

where α and β represent, respectively, the frequency of requests that belong to the organization, and the requests that do not belong to the organization, These requests are classified using the Naive Bayes classifier. k represents the priority level of the space C_2 with respect to the space C_1, where the value of factor k takes the scale from 0.1 to 0.9 inspired from the work presented in [14]. The meaning of the scale is presented in Table 1 (Note: The values 0.2, 0.4, and 0.8 of the factor k mean that the importance of the space C_1 in respect to C_2 is situated between the two adjacent levels shown in the Table 1).

Table 1. Meaning of the scale from 0.1 to 0.9.

Value	Meaning
0.1	The data in the C_1 space have an obvious importance in the network compared to the C_2 space.
0.3	The data in the C_1 space have a higher importance in the network compared to the C_2 space.
0.6	The data in the C_1 space have a low importance in the network compared to the C_2 space.
0.9	The data in the C_1 space have the same importance in the network compared to the C_2 space.

The Lagrangian function associated with this program is written as shown below:

$$L(C_1, C_2, \lambda) = -\alpha C_1 - \beta C_2^k - \lambda(C_1 + C_2 - \text{Max } C) \tag{8}$$

where λ is called the Lagrange multiplier associated with the constraint.

We obtain the desired spaces by calculating the derivatives of our variable as follows:

$$
\begin{cases}
\frac{\partial L}{\partial C_1} = 0 \Leftrightarrow -\alpha - \lambda = 0 \\
\frac{\partial L}{\partial C_2} = 0 \Leftrightarrow -\beta k C_2^{k-1} - \lambda = 0 \\
\frac{\partial L}{\partial \lambda} = 0 \Leftrightarrow -C_1 - C_2 + \text{Max}\, C = 0
\end{cases}
\tag{9}
$$

Finally we obtain the two spaces C_1 and C_2, where C_1 is equal to $\left\lceil \text{Max}\, C - \left(\frac{\alpha}{k\beta} \right)^{\frac{1}{k-1}} \right\rceil$, and C_2 is equal to $\left\lfloor \left(\frac{\alpha}{k\beta} \right)^{\frac{1}{k-1}} \right\rfloor$. We can add the Laplace smoothing to the spaces to avoid zero in the denominator, so C_1 and C_2 will equal, respectively:

$$
\begin{cases}
C_1 = \left\lceil \text{Max}\, C - \left(\frac{\alpha+1}{k\beta+1} \right)^{\frac{1}{k-1}} \right\rceil \\
C_2 = \left\lfloor \left(\frac{\alpha+1}{k\beta+1} \right)^{\frac{1}{k-1}} \right\rfloor
\end{cases}
\tag{10}
$$

In certain cases when the margin between the frequencies of requests belonging to the two classes is very large, we can find that C_1 is equal to a negative number to satisfy our constraint. Since the Lagrange is limited by the constraints of equality, we can fix this problem by affecting $C_1 = 0$, and $C_2 = MaxC$ when $C_1 \leqslant 0$.

4.4. Placement and Replacement of Data

Unlike strategies that share the same space for all content, our strategy allocates the desired space for each content class. It is known that content placement and replacement in NDN is also an important feature that can improve the overall network performance. As shown in Figure 3, when the routers in the network receive the content, they classify it using the classification model. Then, the received content is placed in the corresponding space that was allocated to the same class. If the corresponding space is full, a replacement strategy will be applied to insert the received content by evicting another from the same class. We note that the design of our strategy allows us to adapt any replacement strategy to replace data of the same class, but for simplicity we use the Least Recently Used (LRU) method to replace the received data. In this strategy, the received data will replace the least recently used data [19].
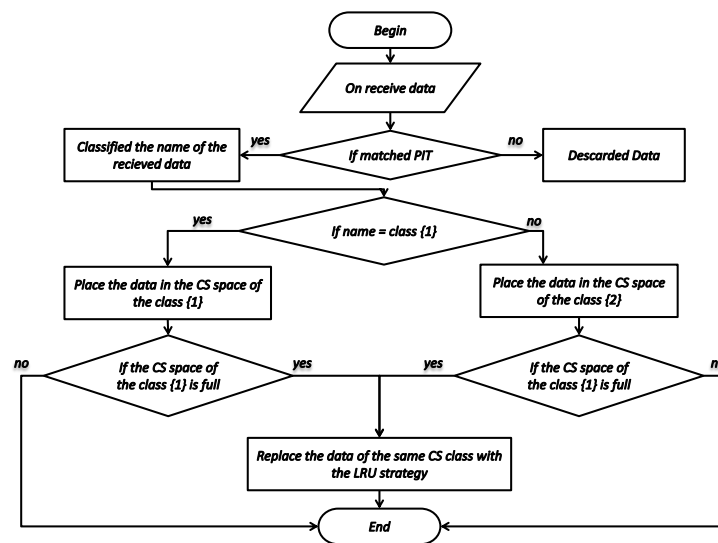

Figure 3. Data placement and replacement process.

5. Performance Results of RACD Strategy

In this section we present and discuss the different performance results obtained. To this end we start by detailing the simulation environment used, and next we proceed to show the actual results achieved. Further, for the sake of reproducibility, the detailed working code of RACD is available at: https://github.com/Herouala/RADC.git (accessed on 12 January 2022).

5.1. Simulation Environment

The performance evaluation of the proposed strategy was performed in two steps. First, we implemented the classification model to classify the dataset used in the simulation. Then, the proposed strategy was evaluated through simulation. Due to the lack of standard datasets in NDN, we constructed our dataset by combining the DMOZ (DMOZ: https://web.archive.org/web/20170312160530/http://rdf.dmoz.org/rdf/content.rdf.u8.gz (accessed on 12 January 2022)) dataset with links extracted from the website of the University Amar Telidji Laghouat (UATL). DMOZ is a well-known dataset for classification, and it is used in many works as in [20,21]. The names of the DMOZ dataset are labeled as data belonging to the class {0} (unimportant), and the links extracted from the UATL are labeled as data belonging to class {1} (important). The dataset used is divided into two parts, one for the training and the other for testing the model. The implemented Multinomial Naïve Bayes classifier model has achieved 98% accuracy in classifying the content names. After that, the RADC strategy is evaluated through ndnSIM [22], which represents the official ns-3-based simulator deployed by the NDN community. ndnSIM enables reliable simulation experiments that can be replicated in real environments without having to modify the source code [23]. To make our simulation as realistic as possible, we chose a real dataset from the University Amar Telidji Laghouat (UATL) as a realistic interest consumption. The UATL dataset contains more than one million links consumed during a week at this university.

The strategies are simulated using a non-complete K-ary tree as in [24–26]. This topology is based on two parameters, D and k, where they represent consecutively the depth of the tree and the number of children of each node in the tree. We chose $D \in [3, 7]$ and k in the interval $[0, 5]$, where k is randomly generated for each node as in [26]. For the simulation scenario, we assume that there are two root nodes, where the first is a server that belongs to the university, and it contains all its data. The second represents a server containing all other data requests. Furthermore, we assume that all requests are sent by leaf nodes.

For each test run, users are assumed to express interests from the used UATL dataset. For simplicity, we set a similar cache size for all nodes in the network. Each link in our topology has a bandwidth of 10 Gbps, which is greater than the traffic demand, and a propagation delay of 1 ms. Table 2 shows the choice of the main parameters for our simulation.

Table 2. Parameter settings for the simulation.

Parameter	Default Value	Range
Tree-ary k	-	[0;5]
Tree depth D	5	[3;7]
node cache capacity	100	[10;100]
Priority level	0.4	[0.1;0.9]
Delay	1 ms	-
Bandwidth	10 Gbps	-

For the evaluation method, we studied the impact of parameters priority level, cache size, and tree depth on the caching performance in the network. We compare the results of our resource allocation mechanism with the mechanism that promotes content sharing in the same cache. The sharing mechanism represents the majority of the proposed strategies,

with a difference in the choice of the placement and replacement strategy. Since RADC does not apply any particular technique to manage the cache content, any strategy could easily be adopted. We compare the two strategies in the same environment, applying the same placement and replacement technique. The two strategies represent, consecutively, LCE [3] and LRU [19] for placement and replacement.

We quantify the performance of the in network caching from two metric aspects:

Cache hit ratio, a metric that measures the ratio between the served content requests and the number of received requests. It also reflects the load saving of the servers, and the delay of the data delivery, since the higher the cache hit ratio is, the faster the data delivery becomes before the user's request reaches the data source. In our case, we study the cache hit of the served data that belong to the university, and also the hit of the other data for both strategies; we can define it as follows:

$$Cache\ hit\ ratio = \frac{r}{|R|} \tag{11}$$

where r represent the number of requests satisfied by in-network caches, and $|R|$ is the total number of requests.

The hop reduction ratio indicates the variation of the data delivery distance in the network. In our study, we investigate the hop reduction ratio of the data belonging to the university against the other data in both tested strategies. The hop reduction ratio is defined as follows:

$$Hop\ reduction\ ratio = 1 - \frac{\sum_{r \in R} h_r}{\sum_{r \in R} H_r} \tag{12}$$

where h_r and H_r are the hop counts from the requester of $r \in R$ to the node, which serves the request, and to the original content server, respectively.

5.2. Performance Results

In the comparison of results, we named the results obtained for the two classes with our strategy as UD-RA and OD-RA, where the UD-RA acronym refers to the results obtained for the data belonging to the university using our strategy, and the OD-RA acronym refers to the obtained results for the data outside the university using our strategy. Further, for the sharing strategy, we name the results obtained for the two classes as UD-sharing and OD-sharing.

5.2.1. The Impact of the Priority Level on Performance

Figure 4 shows the effect of the priority level (k) on the performance of the resource allocation strategy in terms of hit ratio and hop reduction ratio. We can see that, when the priority level k increases, the results of the university data (UD-RA) decrease, and the results of the other data (OD-RA) increase. Here we observe that, from 0.1 to 0.4, there is an increase in the hit ratio and in the hop reduction ratio for university data compared to non-university affiliated data. The reason is that, as the priority coefficient increases, the space given to non-university data increase until it reaches nearly the same size for both types of data when $k = 0.4$. Starting from $k = 0.5$, We can see a clear superiority for OD-RA reaches almost 90% as a difference in the hit ratio, and also for the hop reduction ratio. Notice that, as the k coefficient increases, the space allocated to OD-RA also increases. The reason why OD-RA takes more storage space compared to UD-RA even with the priority coefficient k in the ranges from 0.5 to 0.9 is because there is a large difference between the amounts of data consumed in the two classes, as the amount of data that belong to the university represent only 4% of the total resource consumption. This, in turn, has an impact on space allocation.
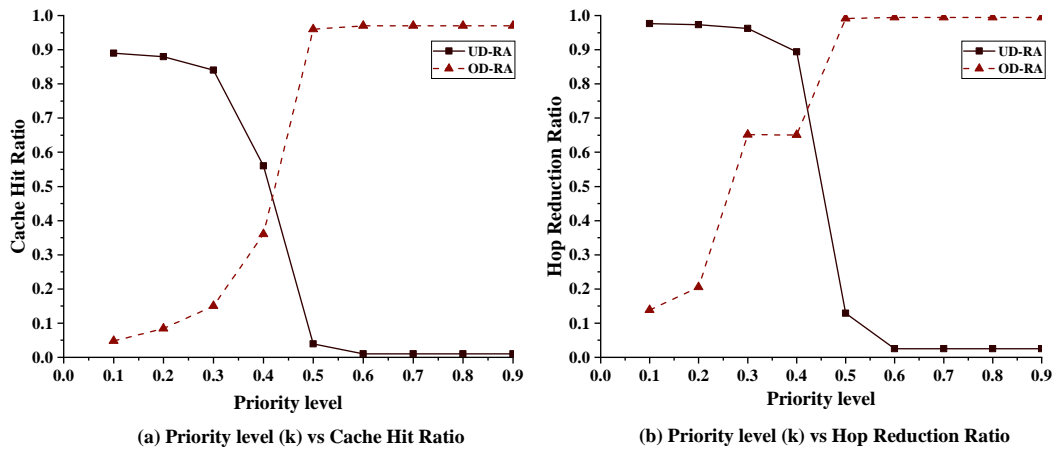
(a) Priority level (k) vs Cache Hit Ratio　　　　　　**(b) Priority level (k) vs Hop Reduction Ratio**

Figure 4. Caching performance vs. priority level in terms of (**a**) cache hit ratio, and (**b**) hop reduction ratio.

5.2.2. Impact of the Cache Size on the Performance of Our Strategy

Figures 5 and 6 show the effect of cache size on the performance of the data sharing and data resource allocation strategy in terms of hit ratio and hop reduction ratio. From the two figures, we can see that, for the sharing strategy, the hop reduction and hit ratio values increase with the cache size, with a large margin for the OD-Sharing results compared to the UD-Sharing. The improvement in results when cache size increases makes sense because now more data are cached, which increases the chances of having the data searched to the closest possible node. The difference in the results between the two classes is due to the fact that the requests in the OD-Sharing class are more numerous than those of UD-Sharing.
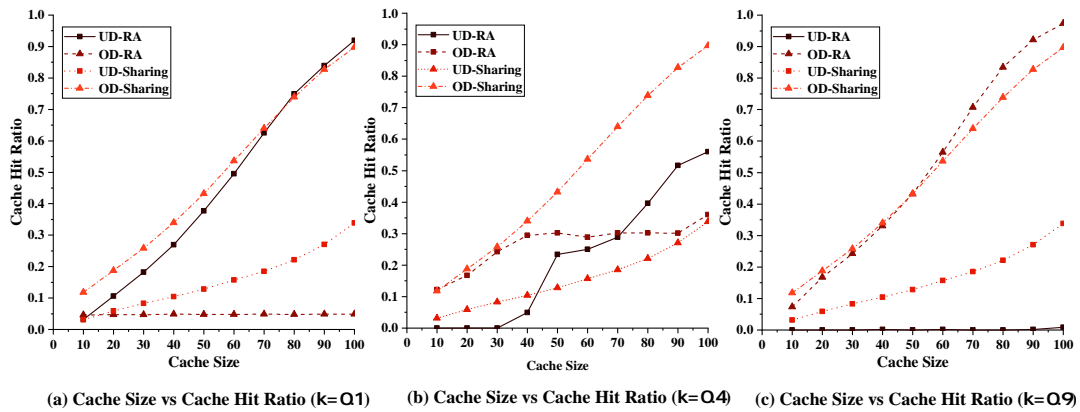

(a) Cache Size vs Cache Hit Ratio (k=0.1)　**(b) Cache Size vs Cache Hit Ratio (k=0.4)**　**(c) Cache Size vs Cache Hit Ratio (k=0.9)**

Figure 5. Caching performance vs. cache size in terms of: cache hit ratio.

(a) Cache Size vs Hop Reduction Ratio (k=0.1)　**(b) Cache Size vs Hop Reduction Ratio (k=0.4)**　**(c) Cache Size vs Hop Reduction Ratio (k=0.9)**
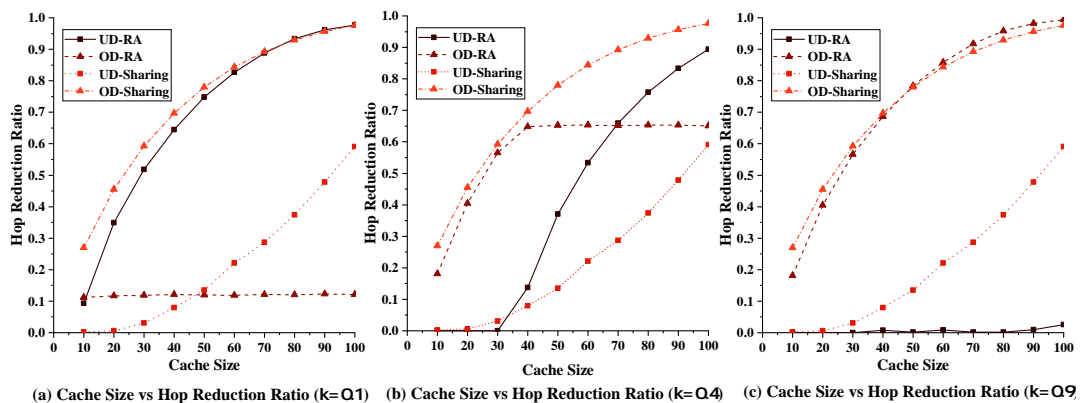
Figure 6. Caching performance vs. cache size in terms of hop reduction ratio when varying *k*.

For the resource allocation strategy, with $k = 0.1$, the university data (UD-RA) achieves a higher hit ratio and hop reduction ratio as the cache size increases. The OD-RA achieves low and constant values compared to UD-RA. This is because, when using a very small value for coefficient k, the cache partition size given to UD-RA is much larger than the size given to OD-RA, which was small and fixed in all sizes tested. When $k = 0.9$, we find that our strategy performs inversely to $k = 0.1$ where, as the cache size increases, OD-RA performs well compared to UD-RA in terms of hit ratio and hop reduction ratio. The reason is that, when the coefficient is $k = 0.9$, the coefficients of the two data classes become almost equivalent; therefore, the allocations of each partition become related to the amount of data received from both classes. Since the consumed university data represents only 4% of the total consumption, all the cache is allocated to OD-RA data. We also notice that, when $k = 0.4$, from 10 to 30, the UD-RA obtains a null hit and hop reduction ratio, in contrast to the OD-RA, which increases with the increase in cache size. From 40 to 60 we notice an increase in the results of both classes of data, with their convergence until they are equal in the capacity 70. From 80 to 100, the results show a continuous increase in both classes, with dominance of UD-RA over OD-RA. The reason for the continuous increase in results for both classes is that both classes obtain more space when the total cache space increases. The reason why these results are obtained when $k = 0.4$ is that, when the cache size is between 10 and 30, the Lagrange utility function gives the total memory space for the OD-RA. Then, beyond size 40, the function starts to give more memory space for UD-RA, and OD-RA space remains fixed until they reach equal results for size 70. Then, the increase in space continues for UD-RA, which takes more memory space, and thus achieves better results than OD-RA.

5.2.3. Impact of Tree Depth on Performance

Figures 7 and 8 show the effect of tree depth on the performance of the sharing and resource allocation strategies by calculating the hit ratio and hop reduction ratio for each strategy. For the sharing strategy, we observe a slight increase in the results for both classes of data in terms of hit and hop reduction ratio as depth increases. We also observe a large difference in the results for the two classes, with a 50% difference in hits, and a difference ranging from 60% to 35% in hop reduction results between OD-Sharing and UD-Sharing. The increase in the results obtained is due to the number of nodes that increase with length, allowing more cache space to be allocated. This allows more data to be cached, including low-consumption data, such as university-owned data (UD-Sharing), which results in a higher hit ratio and less hop ratio.
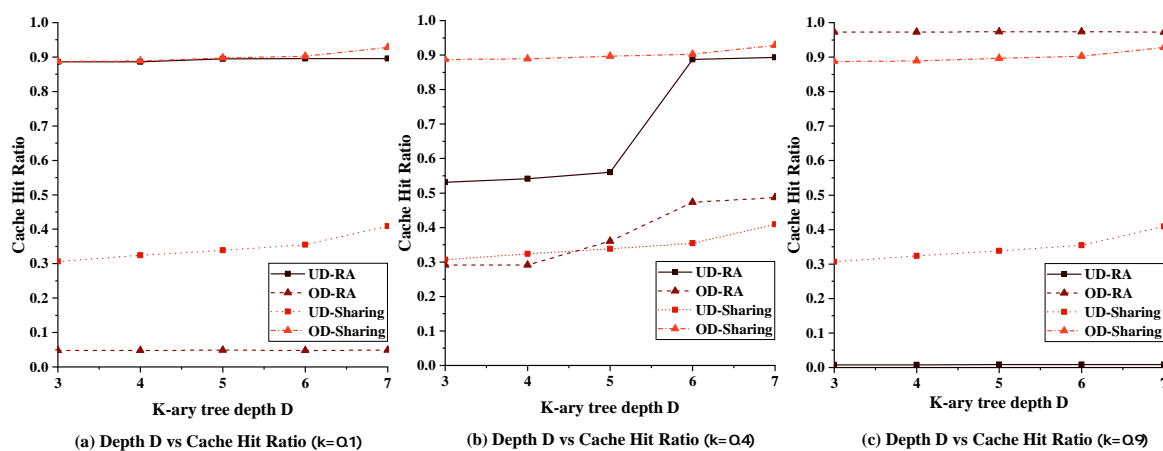

Figure 7. Caching performance vs. tree depth in terms of cache hit ratio when varying k.
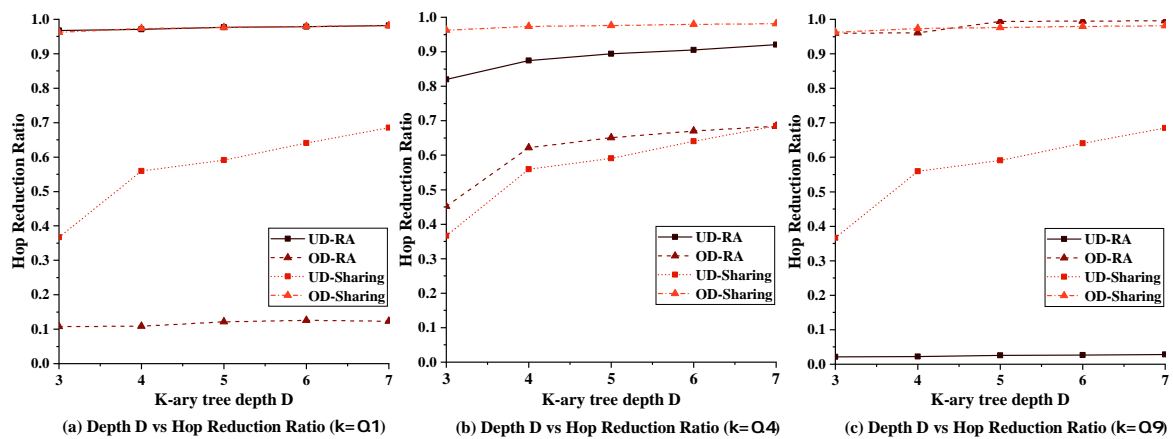
Figure 8. Caching performance vs. tree depth in terms of hop reduction ratio when varying *k*.

Regarding the resource allocation strategy, the results differ depending on the *k* factor. When $k = 0.1$, we notice a very slight increase in the hit and hop reduction results with the increase in depth D. We also notice a big difference between the results obtained for the two classes of data, being that this difference grows up to 85% between the UD-RA and the OD-RA, where OD-RA achieves the lowest percentage. Indeed, the use of a coefficient $k = 0.1$ leads to a very high priority for the data that belong to the university compared to the other data. With the use of $k = 0.9$, we see the opposite of the results given with the use of $k = 0.1$. This is because the coefficients between the two classes of data become closer, and thus the priority becomes dependent on the amount of consumed data for both classes. Since the amount of consumed data that are not affiliated with the university is much larger than the others, OD-RA gained a significant advantage within the given space, leading to the obtained results. At $k = 0.4$, there is an improvement in the results of both classes as the depth D increases. The difference between the two classes of data is between 20% and 60% for the hit ratio, and between 40% and 25% for the hop reduction ratio, with a preference for the UD-RA. The convergence of the results obtained for the two classes is due to the fact that the space given to them is similar, with a bias towards the space reserved for the university data.

Through this study and the results obtained, we can say that the proposed strategy offers the possibility to control the data in an excellent way, even at the levels where there is a big difference between the consumed data classes. Indeed, by using this strategy, we are able to determine the desired priority level, which in turn can allocate space for the desired data resources.

6. Conclusions and Future Work

The proposed strategies for resource allocation in NDNs are known to be effective in optimizing the overall network performance; however, these strategies may suffer in a limited network environment. That is because a fundamental trade-off problem between resource efficiency and user satisfaction might arise. To this end, a resource allocation scheme based on data classification has been proposed with a weighting factor that is used to decide at which level we want to control the aforementioned trade-off. Simulation results show that our strategy outperforms the sharing strategies, since it proves its ability to control any desired trade-off point of their respective efficiency and satisfaction levels.

As future work, further experiments should be performed to evaluate the robustness of our approach against other state-of-the-art approaches. We also want to add other factors to the objective function of the resource allocation which may help at enhancing and improving network performance.

References

1. Vasilakos, A.V.; Li, Z.; Simon, G.; You, W. Information centric network: Research challenges and opportunities. *J. Netw. Comput. Appl.* **2015**, *52*, 1–10.
2. Abdullahi, I.; Arif, S.; Hassan, S. Survey on caching approaches in information centric networking. *J. Netw. Comput. Appl.* **2015**, *56*, 48–59.
3. Zhang, L.; Afanasyev, A.; Burke, J.; Jacobson, V.; Claffy, K.; Crowley, P.; Papadopoulos, C.; Wang, L.; Zhang, B. Named data networking. *ACM SIGCOMM Comput. Commun. Rev.* **2014**, *44*, 66–73.
4. Ahmad, F.; Kerrache, C.A.; Kurugollu, F.; Hussain, R. Realization of blockchain in named data networking-based internet-of-vehicles. *IT Prof.* **2019**, *21*, 41–47.
5. Cisco Annual Internet Report (2018–2023) White Paper. Available online: https://www.cisco.com/c/en/us/solutions/collateral/ executive-perspectives/annual-internet-report/white-paper-c11-741490.html (accessed on 15 January 2022).
6. Kerrche, C.A.; Ahmad, F.; Elhoseny, M.; Adnane, A.; Ahmad, Z.; Nour, B. Internet of vehicles over named data networking: current status and future challenges. In *Emerging Technologies for Connected Internet of Vehicles and Intelligent Transportation System Networks*; Springer: Cham, Switzerland, 2020; pp. 83–99.
7. Psaras, I.; Chai, W.K.; Pavlou, G. In-network cache management and resource allocation for information-centric networks. *IEEE Trans. Parallel Distrib. Syst.* **2013**, *25*, 2920–2931.
8. Yao, J.; Yin, B.; Lu, X. A novel joint adaptive forwarding and resource allocation strategy for named data networking based on SMDP. In Proceedings of the 2016 12th IEEE International Conference on Control and Automation (ICCA), Kathmandu, Nepal, 1–3 June 2016; pp. 956–961.
9. Li, C.; Xie, R.; Huang, T.; Huo, R.; Liu, J.; Liu, Y. Joint Forwarding Strategy and Resource Allocation in Information-Centric HWNs. In Proceedings of the GLOBECOM 2017-2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6.
10. Yuan, D.; Xu, Y.; Ran, J.; Hu, H.; Liu, Y.; Li, X. An optimal fair resource allocation strategy for a lightweight content-centric networking architecture. In Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 13–16 December 2017; pp. 573–577.
11. Zhang, Y.; Tan, X.; Li, W. In-network cache size allocation for video streaming on named data networking. In Proceedings of the 2017 VI International Conference on Network, Communication and Computing, Kunming, China, 8–10 December 2017; pp. 18–23.
12. Ehsanpour, M.; Bayat, S.; Hemmatyar, A.M.A. On Efficient and Social-Aware Object Allocation in Named Data Networks Using Matching Theory. In Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC), Natal, Brazil, 25–28 June 2018; pp. 298–303.
13. Zhang, M.; Xie, P.; Zhu, J.; Wu, Q.; Zheng, R.; Zhang, H. NCPP-based caching and NUR-based resource allocation for information-centric networking. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 1739–1745.
14. Huo, R.; Xie, R.; Zhang, H.; Huang, T.; Liu, Y. What to cache: differentiated caching resource allocation and management in information-centric networking. *China Commun.* **2016**, *13*, 261–276.
15. Yovita, L.V.; Syambas, N.R.; Edward, I.Y.M. CAPIC: Cache based on popularity and class in named data network. In Proceedings of the 2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), Bandung, Indonesia, 5–7 December 2018; pp. 24–29.
16. Naeem, M.A.; Nor, S.A.; Hassan, S.; Kim, B.S. Compound popular content caching strategy in named data networking. *Electronics* **2019**, *8*, 771.
17. McCallum, A.; Nigam, K. A comparison of event models for naive bayes text classification. In Proceedings of the AAAI-98 Workshop, Madison, WI, USA, 26–27 July 1998; Volume 752, pp. 41–48.
18. Dehghan, M.; Massoulie, L.; Towsley, D.; Menasche, D.S.; Tay, Y.C. A utility optimization approach to network cache design. *IEEE/ACM Trans. Netw.* **2019**, *27*, 1013–1027.
19. Situmorang, H.; Syambas, N.R.; Juhana, T.; Edward, I.Y.M. A Simulation of Cache Replacement Strategy on Named Data Network. In Proceedings of the 2018 12th International Conference on Telecommunication Systems, Services, and Applications (TSSA), Yogyakarta, Indonesia, 4–5 October 2018; pp. 1–4.

20. Kim, J.; Ko, M.C.; Kim, J.; Shin, M.S. Route Prefix Caching Using Bloom Filters in Named Data Networking. *Appl. Sci.* **2020**, *10*, 2226.

21. Zaeem, R.N.; Barber, K.S. A Large Publicly Available Corpus of Website Privacy Policies Based on DMOZ. In Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy, Virtual, 26–28 April 2021.

22. Mastorakis, S.; Afanasyev, A.; Zhang, L. On the evolution of ndnSIM: An open-source simulator for NDN experimentation. *ACM SIGCOMM Comput. Commun. Rev.* **2017**, *47*, 19–33.

23. Amadeo, M.; Ruggeri, G.; Campolo, C.; Molinaro, A. IoT services allocation at the edge via named data networking: From optimal bounds to practical design. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 661–674.

24. Chai, W.K.; He, D.; Psaras, I.; Pavlou, G. Cache "less for more" in information-centric networks (extended version). *Comput. Commun.* **2013**, *36*, 758–770.

25. Ren, J.; Qi, W.; Westphal, C.; Wang, J.; Lu, K.; Liu, S.; Wang, S. Magic: A distributed max-gain in-network caching strategy in information-centric networks. In Proceedings of the 2014 IEEE conference on computer communications workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 27 April–2 May 2014; pp. 470–475.

26. Hu, X.; Gong, J.; Cheng, G.; Fan, C. Enhancing in-network caching by coupling cache placement, replacement and location. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015; pp. 5672–5678.