



Randomized SVD Based Probabilistic Caching Strategy in Named Data Networks

Narges Mehran^{a,*}, Naser Movahhedinia^a

^a*Department of Computer Architecture, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran.*

ARTICLE INFO.

Article history:

Received: 29 November 2016

Revised: 22 September 2017

Accepted: 06 November 2017

Published Online: 01 April 2018

Keywords:

Named Data Network, Caching Strategy, Betweenness Centrality, Linear Weighted Moving Average, Randomized-SVD

ABSTRACT

Named Data Networking (NDN), a data-centric enabled-cache architecture, as one of the candidates for the future Internet, has the potential to overcome many of the current Internet difficulties (*e.g.*, security, mobility, multicasting). Influenced by using cache in intermediate equipment, NDN has gained attention as a prominent method of Internet content sharing. Managing the NDN caches and reducing the cache redundancy are the important goals in this paper. Our main contribution in this research is toward caching optimization in comparison with betweenness probabilistic in-network caching strategy. Therefore, with respect to combined impacts of long-term centrality-based metric and Linear Weighted Moving Average (LWMA) of short-term parameters such as user incoming pending requests and unique outgoing hit requests on caching management, a flexible probability caching strategy is proposed. Moreover, a simple Randomized-SVD approach is applied to combine averaged short-term and long-term metrics. The output of this data-fusion algorithm is used to allocate a proper probability to the caching strategy. Evaluation results display an increase in the hit ratios of NDN routers' content-stores for the proposed method. In addition, the producer's hit ratio and the Interest-Data Round Trip Time, compared to the betweenness scheme, is decreased.

© 2016 JComSec. All rights reserved.

1 Introduction

According to the Visual Networking Index (VNI) Global Mobile Data Forecast [1], monthly global mobile data traffic will be 49 Exabyte of which 78 percent will be video contents by 2021. To handle this increasing traffic volume of online contents, an emerging paradigm has been developed for future Internet architecture, called Information-Centric Network (ICN) [2] supporting in-network caching to efficiently transport named sub-objects (or chunks) to users.

* Corresponding author.

Email addresses: n.mehran@eng.ui.ac.ir (N. Mehran), naserm@eng.ui.ac.ir (N. Movahhedinia)

ISSN: 2322-4460 © 2016 JComSec. All rights reserved.

Content-Centric Networking project [3] (recently known as Named Data Networking (NDN) [4]) is one of ICN architectures proposed by PARC (Palo Alto Research Center, formerly Xerox PARC) in 2009. The goal of designing this architecture is to perform management, routing and resource allocation according to the content ID, instead of the host ID.

As ubiquitous caching [4] increases the costs of memory consumption, unnecessary caching operations may be reduced by probabilistic caching in the nodes. In this research, a novel metric for caching strategy in NDN networks is examined and evaluated based on the importance of the NDN routers. The proposed metric encompasses some short-term parameters, such as



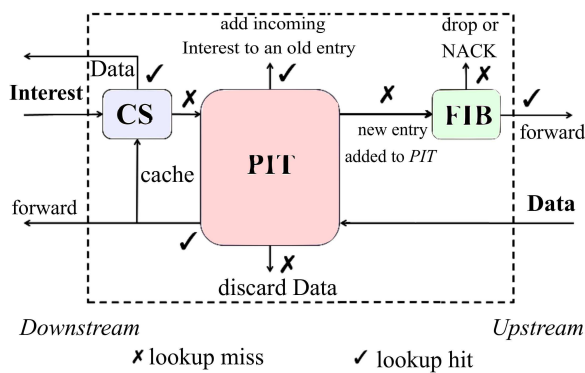


Figure 1. Interest and Data Processing in an NDN Router [3].

unique outgoing traffic distribution and users' pending incoming requests, as well as a long-term one namely topology-related betweenness-centrality of the nodes. We modify the metrics used by NDN cache size allocation schemes proposed in [5–7], and the Adaptive Congestion Control Protocol in NDN (ACCP_{ndn}) presented in [8], to obtain a more general model for a probabilistic caching decision applied to NDN routers. These parameters help to come across a node's importance with regard to its caching duty in NDN network. As the former two parameters may change frequently, they are smoothed out by Linear Weighted Moving Average scheme in a proper time interval. Next, the results are combined with the betweenness values of the routers to encounter topological information. Eventually, to denoise the gathered data and reduce the computational complexity of the algorithm, the important features are extracted by a Randomized-SVD method.

In Section 2, we firstly describe the technology considered for future ICN architectures and the NDN routers and then focus on data-fusion algorithms as the basis of the proposed method. Section 3 reviews the related works presented specifically on NDN strategies for storage devices. In Section 4, our main contribution is discussed. Section 5 is devoted to evaluations of the proposed method. In Section 6, the paper is finally concluded.

2 Background

2.1 Information Centric Networking

Nowadays, Peer-to-Peer (P2P) and Content Distribution Networks (CDNs) are two common broad content-centric Internet technologies. Content providers can purchase the resources from CDN operators owning a set of geographic replicated machines. This is a highly beneficial point for content providers as their content would be available on the worldwide Internet. Likewise, in P2P networks, users contribute a share to the resources and receive content-centric services in

return. All of these activities are organized in overlay layers; so that the core routers are not aware of requested contents [9].

The pioneer works proposing a redefinition of the Future Internet architecture have focused on mapping content names to locations by in-network caching employment. ICN is a notable concept to organize an infrastructure with the help of which an enhanced support will be provided for content distribution and mobility [10]. It focuses on finding and delivering named contents, instead of maintaining end-to-end communications between hosts identified by IP addresses [11]. Among these, in this research, NDN architecture, thanks to CCNx [12] and NDN [13] projects with their simple, robust and effective communication models, has been selected.

2.2 NDN in a Nutshell

NDN proposes a communication standard based on the exchange of two types of packets: Interest and Data that carry Uniform Resource Identifier (URI)-like content names. A consumer requests a named content by broadcasting an Interest packet over its available network faces. The Interest is forwarded hop-by-hop until a producer provides the requested Data. As noticed in Figure 1, an NDN router has three data structures to cache and forward the Interest and Data packets: Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB).

CS, as the name implies, with defined placement and replacement policies, stores chunks of incoming Data messages for satisfying future possible Interests (requested by a user) in its storage device, simply named cache. PIT stores request not satisfied by CS and passed to next router. Finally, FIB forwards Interest messages to the proper face of an NDN router. When the content's location is broadcasted to all the NDN routers, it will be populated into FIBs. On the way that Data message is forwarding back to its destination, if an entry in PIT is matched, the data chunk is immediately forwarded by to its destination [4].

2.3 Data-Fusion Algorithms

To store, analyze and summarize the vast amounts of generated data, one may reduce the dimension of data by dimensionality-reduction data fusion. This transformation finds a subspace whose vectors are a combination of the old subspace and projects a t -dimensional space onto an f -dimensional subspace of the original features, where $f < t$. There are several learning-based dimensionality-reduction data-fusion schemes, like ISOMAP, but these methods are slow and also impose a high computational load on pro-



cessing the input data, especially for large-scale real-network parameters. The learning-based clustering algorithms, such as K-Means [14], need heavily to have some initial values to randomly group data points into clusters. As a result, the raw data sets collected from our implementations are not available before running the algorithm. For reducing the size of information and combining features with different qualities, Truncated-SVD is exploited; according to [15], Truncated-SVD has the ability to extract only the most important information from the data matrix by using just the first several components estimated from the original matrix of data set. Randomized-SVD implements a type of Truncated Singular Value Decomposition (Truncated-SVD) that only computes the k largest singular values with a randomized algorithm, where k is a user-specified parameter [16].

Randomized-SVD is similar to PCA [17], but differs in that it works on sample matrices X directly instead of their covariance matrices. When the column-wise (per-feature) means of X is subtracted from the feature values, Randomized-SVD on the resulting matrix is equivalent to PCA [15].

Given an $m \times n$ matrix X , a target number k of singular vectors, this procedure computes an approximate factorization $U\Sigma V^*$, where U and V are orthonormal matrices whose columns are eigenvectors of $X.X^*$ and $X^*.X$ respectively, and Σ is nonnegative and diagonal matrix containing the eigenvalues of $X.X^*$ in the diagonal being sorted in descending order. As illustrated in [16], by considering the problem of finding the k principal components of the SVD of an $m \times n$ input matrix, randomized algorithms involve $O(mn \log(k))$ floating-point operations (flops) in distinction to $O(mnk)$ for classical algorithms.

The six-step implementation of Randomized-SVD, briefly shown in the following pseudo code (Algorithm 1), can generate a structure from an unstructured input data matrix by using a subsampled random Fourier Transform (SRFT) and QR decomposition:

Algorithm 1 Randomized-SVD's Pseudo Code [18]

GOAL: GIVEN AN $m \times n$ INPUT MATRIX X , COMPUTE AN APPROXIMATE RANK- k SVD: $X \approx U_k \cdot \Sigma_k \cdot V_k^T$

- 1: Draw an $n \times k$ Gaussian random matrix Ω ,
 - 2: Form an $m \times k$ orthonormal matrix Q by using (subsampled) FFT and QR factorization,
 - 3: Form the $k \times n$ matrix $B = Q^T \cdot X$
 - 4: Compute the SVD of the small matrix B : $B = \tilde{U} \cdot \Sigma_k \cdot V_k^T$,
 - 5: Form the matrix $U_k = Q \cdot \tilde{U}$,
 - 6: Calculate $X_k = U_k \cdot \Sigma_k \cdot V_k^T$.
-

3 Related Works

In [19], authors have presented a scheme called Autonomous System Collaboration Caching Strategy (ASCCS). In ASCCS scheme, CCN network is divided into several Autonomous Systems (AS) as OSPF does, then the control nodes are selected according to the betweenness and the cache replacement rate of the nodes. Each AS is centralized, and controlled by the control node. Then the selected control node is responsible for computing the popularity of different contents to define the suitable cache policy for them. Therefore, routers in the same AS collaborate with each other to improve the performance of the network caches.

In [20], the authors have been concerned with cache management operations and cache network modeling that have to be adjusted to fit in a completely decentralized environment. They have focused on the allocation of the available cache capacity along a request-response path; however, it is just for a hierarchical network topology. In other words, authors have suggested that if ICN routers are allocated the caching probability, associated with the path capacity, they achieve better network performance.

PopCache [21] allows an ICN router to store the content in accordance with the popularity characteristic of it. The closest router to the consumers store the content with the most popularity and the less popular contents are cached in other routers. In addition, the authors have taken into account the rule of distribution of caching data packets along the path. They have indicated that in terms of Round-Trip Time (RTT), PopCache has a lower expected round-trip time than Ref. [20], especially when the Zipf exponent becomes larger.

In [22], authors have proposed a centrality-based caching algorithm by exploiting the concept of betweenness-centrality to improve the caching gain. The possibility of caching less in order to achieve higher performance gain has been investigated. They have also argued that the universal caching strategy is costly and sub-optimal.

Authors in [23] present a Cost-Aware cache decision policy, contrary to classic strategies, that cooperatively consider content popularity and economic aspects of links through which the contents have to be retrieved. It is shown that ICN can help ISPs to reduce the cost related to inter-domain traffic. Cost-Aware policies just allow the contents to be cached that are costly for the operator because they reduce the load on expensive and cheap links while increasing the load on free links. Due to their evaluations, by tuning the tradeoff between popularity and price, the cost-aware performance is classified into mostly popularity-driven,



balanced, and mostly cost-driven categories. A larger gain is achieved in balanced mode, since designing a better composition of hit ratio and cost fraction.

4 Problem Definition and Proposed Scheme

4.1 Problem Definition

The problem inspected in this article is how to decide on a probabilistic placement strategy to improve NDN performance in terms of power consumption of wasteful caching while preserving the cache hits and low end-to-end delay. Therefore, we present a scheme focusing on the content store structure of an NDN router.

After collecting short-term dynamic parameters, and combining them with a long-term static parameter, the appropriate probabilistic caching strategy is derived. Indeed, each router is allowed to cache just based on this controllable probability. The important long-term centrality metric is betweenness; showing how many times, a router is on the shortest path between Producer-Consumer pairs. As the data chunks would be routed for the requester based on the PIT table, more interfaces stored in a PIT may cause more data chunks expecting to be received. Moreover, more numbers of cache hits actually show more Interests satisfied by the cache. Altogether, the Unique Hit Interests along with the users' Pending Interests in a router within an appropriate time interval, in conjunction with the betweenness-centrality of the router, are utilized to calculate the caching probability of an NDN router.

4.2 Metric Definition

If NDN is considered as a graph, the routers and the point-to-point links between routers represent the nodes and the links in the graph correspondingly. As the links are assumed symmetric and full duplex, the equivalent graph would be unidirectional. To find out how influential a node in a group is, a graph-related long-term parameter, betweenness-centrality (BC), precomputed offline, is used. This metric is an indicator of node centrality in the graph of a network. In order to find this type of centrality, the Equation (1) to Equation (3) are adopted [24].

The pair-wise dependencies of two nodes, s and t , on a special node, v , is calculated by Equation (1) [24]:

$$\delta(s, t|v) = \frac{\sigma(s, t|v)}{\sigma(s, t)} \quad (1)$$

where $\sigma(s, t)$ shows the number of different shortest paths from node s to t . In other words, the pair-

wise dependency shows the fraction of all the number of geodesics between the pair of nodes, $\sigma(s, t)$, that include node v , $\sigma(s, t|v)$.

By aggregating these computations, we can have Equation (2) [24]:

$$\delta(s|v) = \sum_{t \in V} \delta(s, t|v) \quad (2)$$

Then, for all $s, v \in V$, we can compute the one-sided dependency of s on vertices, one edge farther away [24]:

$$\delta(s|v) = \sum_{w: \left\{ \begin{array}{l} (v, w) \in E \text{ and} \\ \text{dist}(s, w) = \text{dist}(s, v) + 1 \end{array} \right\}} \frac{\sigma(s, v)}{\sigma(s, w)} \cdot (1 + \delta(s|w)) \quad (3)$$

The next phase is to collect the information including the number of stored pending interests and the number of unique data chunks being satisfied by the current node. These two metrics are sampled at the NDN routers. As the entries in PIT table are unique, the Hit Interests, satisfied by different data chunks in the current router, are also adopted as another dynamic metric. In order to approximate and smooth the fluctuations of the obtained samples, a type of Weighted Moving Average (WMA) [25] procedure is applied because of its efficiency and simplicity. In this WMA model, named Linear WMA (LWMA) [26], the current and the several past samples are linearly weighted and added together. In other words, we can further predict the samples based on the behavior of their past. More specifically, the data set constitutes a time series for each router. By applying a time-series prediction method to the data set, activity of a sample at time t can be forecasted. In this case, from the n total number of samples, just m number of them are chosen as a window for the prediction, where $m \ll n$; in calculating ratio of the weights, the denominator is equal to the addition of all the weights and the numerator is selected from the window of LWMA weights.

Therefore, we declare Equations (4) and (5) to compute the weights in an m -size moving window for both the Pending Interest and Unique Hit Interest samples. In calculating the ratio of the weights, the denominator is equal to the addition of all the weights and the numerator is selected from the window as follows:

$$PI_Weight(w) = \frac{(m - (w - 1))}{m(m + 1)/2}; w \in [1, m] \quad (4)$$

$$UniqHI_Weight(w) = \frac{(m - (w - 1))}{m(m + 1)/2}; w \in [1, m] \quad (5)$$

In this way, LWMA requires a forgetting factor; as



the older the observation, the less the weight taken. Therefore, the smallest weight is allocated to the oldest sample, and it is continued until the largest weight is allocated to the newest sample in the window. The LWMA of PI and Unique HI samples are attained as Equations (6) and (7):

$$(New_PI_Sample * PI_Weight)[t] = \sum_{w=1}^t \left\{ New_PI_Sample(w) \times PI_Weight(t - (w - 1)) \right\}; t \in [1, n] \quad (6)$$

$$(New_UniqHI_Sample * UniqHI_Weight)[t] = \sum_{w=1}^t \left\{ New_UniqHI_Sample(w) \times UniqHI_Weight(t - (w - 1)) \right\}; t \in [1, n] \quad (7)$$

where $New_PI_Sample(t)$ is the time series of PIT snapshots and $New_UniqHI_Sample(t)$ is the time series of unique CS hit events. The sampling frequency should be selected based on the behavior of fluctuations; therefore, ten samples per second (every 0.1 second) is an almost fair one.

In Section 4.3, we will describe how to use a Randomized-SVD method to allocate a probability to router's cache, based on these time-variant variables and time-invariant one.

4.3 The Proposed Caching Strategy

Estimating the LWMA of unique outgoing Hit Interests and users' Pending Interests, the Randomized-SVD data fusion algorithm is applied to merge the dynamic parameters with static betweenness-centrality one. This algorithm extricates the most important correlated information along with some orthogonal uncorrelated factors; *e.g.*, if X is the observations from two features, then it is interpreted as a geometrical representation of the two original axes. In large-scale data sets, it can be seen as an $m \times n$ matrix, where the value m is the number of routers in the network, and n is the number of features collected from every router. Hence, this is the input to our Randomized-SVD algorithm, and the output is a vector with m elements showing a value for a router. In our work, this value implies the importance of a router in NDN. The steps are briefly as follows:

At first, normalizing the data (Equation (8)),

$$x = \frac{x - mean}{std_dev} \quad (8)$$

Second, Randomized-SVD, applied to the training samples X , produces a low-rank approximation X_k , as defined in Equation (9):

$$X \approx X_k = U_k \cdot \Sigma_k \cdot V_k^T \quad (9)$$

where matrices U_k and V_k , with orthonormal columns, include the normalized eigenvectors of $X_k^T \cdot X_k$ and $X_k \cdot X_k^T$ respectively. The nonnegative diagonal matrix Σ_k includes the same eigenvalues of matrix $X_k \cdot X_k^T$ and $X_k^T \cdot X_k$.

$U_k \Sigma_k$ is a transformed training set with k features. To also transform the original set X , we multiply it with V_k (the normalized eigenvectors of a new subspace), as shown in Equation (10):

$$X_{fused} = X \cdot V_k \quad (10)$$

Having the raw data table processed by Randomized-SVD data-fusion algorithm, *i.e.*, the elements of vector V in Equation (10), then vector X_{fused} is obtained for each router. At the end, the elements of vector X_{fused} are normalized to calculate the caching probability of the router.

A flow diagram of the above data fusion process is depicted in Figure 2. The variables of NDN router i are denoted by: $X_1[i]$, $X_2[i]$, $X_3[i]$, and the components of first eigenvector are: $Component_1$, $Component_2$ and $Component_3$, as observed in Equation (11):

$$Component_1 + Component_2 + Component_3 = 1 \quad (11)$$

Then, the fused data for the content router i is obtained by Equation (12):

$$X_{fused}[i] = \left\{ \begin{array}{l} X_1[i]Component_1 + \\ X_2[i]Component_2 + \\ X_3[i]Component_3 \end{array} \right\} \quad (12)$$

Using this value, the percentage of a cache or the probability is calculated. By that probability, a cache is allowed to store data chunks.

Algorithm 2 displays the proposed scheme. In this algorithm, all the steps are computed offline except the second step, needing the runtime of NDN. Firstly, the centrality metric of every router is computed by the router's role in the NDN graph. Afterward, by running the NDN network, the short-term parameters are calculated. Then, the eigenvalues of the short-term metrics are estimated. In the fourth and fifth steps, the components of the reduced data set are calculated from the three mentioned parameters. At last, normalizing the fused data, the importance of routers and consequently the cache probability is obtained.



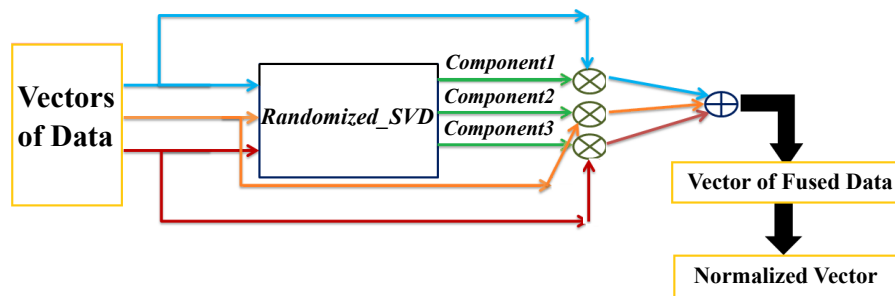


Figure 2. The Data Fusion Diagram of Proposed Method.

Algorithm 2 Proposed Cache Size Allocation Algorithm

PROPOSEDCACHINGSTRATEGY (*Topology of NDN Network*)

- 1: Calculating betweenness-centrality of every node offline,
 - 2: **for** every NDN router **do**
 - 3: Counting the number of PIs in PIT and the number of Unique HIs in current CS,
 - 4: **end for**
 - 5: Estimating the LWMA of counters in previous steps,
 - 6: Apply the data-fusion Randomized-SVD algorithm on vectors of: (Betweenness, LWMA of PIs, LWMA of UniqHIs),
 - 7: Using the components to calculate the fused data for every router,
 - 8: After normalizing the fused-data vector, the importance of the NDN router is presented,
 - 9: Computing the normalized caching probability, based on the NDN router's importance.
-

By neglecting the additional cost of online runtime, the most complex parts of this algorithm are steps 1 and 3. The algorithm for Betweenness-centrality metric of every router runs in $O(NM)$ time on unweighted networks [24], where M is the number of links and N is the number of routers in the NDN network. After gathering the information from step 2, the load of processing them in step 3 is about $O(Nt_s)$, where N is the number of NDN core routers and t_s is the number of samples of PIs and Unique HIs. In step 4, by having just three features from an NDN router, it takes $O(kN)$ to compute the components. k is the number of feature. Other steps also takes $O(N)$ to calculate the weights as an importance for every NDN cache. Therefore, the total time complexity of Algorithm 2 is $O(NM + Nt_s + kN)$, which is a polynomial time complexity. In other words, the complexity of this algorithm is dependent on network topology and the number of samples from all the NDN routers' short-term metrics.

5 Evaluations

NDN, amongst ICN architectures, is a more complete promising Future Internet architecture, including a software development tool (CCNx [12]), and simulation softwares like ndnSIM [27] and ccnSim [28]. In addition, a new ICN simulator, Icarus [29], has been developed. Icarus offers researchers to evaluate their studies under other ICN architectures [30]. However,

to cope with our future needs, our focus is on an NDN based simulator that can be extended to other TCP/IP protocols beside the proposed method. As such, we evaluate our proposed method using the open source NS-3 [31] based simulator, ndnSIM-v2.1 released by the Internet Research Lab of UCLA.

The experiments are performed on a real-network topology, named Abilene. Abilene is an advanced backbone network operated by the University Corporation for Advanced Internet Development (UCAID) to support the physical components of the Internet2 project [32]. This topology has been used in various assessments for the CCN network, including a CCN-based channel zapping protocol [33]. Our experiments are tested on a scenario as observed in Figure 3. The NDN routers are connected by point-to-point links with a bandwidth of 10Gbps. Twelve consumer nodes, attached to the NDN routers, arbitrarily extend the topology. The ndn::BestRoute [34] Forwarding Strategy, a module defining how Interest and Data are being forwarded, is selected. The chunk size in NDN is expected to be a packet-level size such as 1KB [35]. Table 1 displays the parameters considered in simulations. The cache replacement algorithm is Least Recently Used (LRU). The chunk-based cache size is set to 1040000 chunks. Assuming Mandelbrot-Zipf distribution for content retrievals by users, as shown in Equation (13), a content distribution of P2P network is modeled [36].



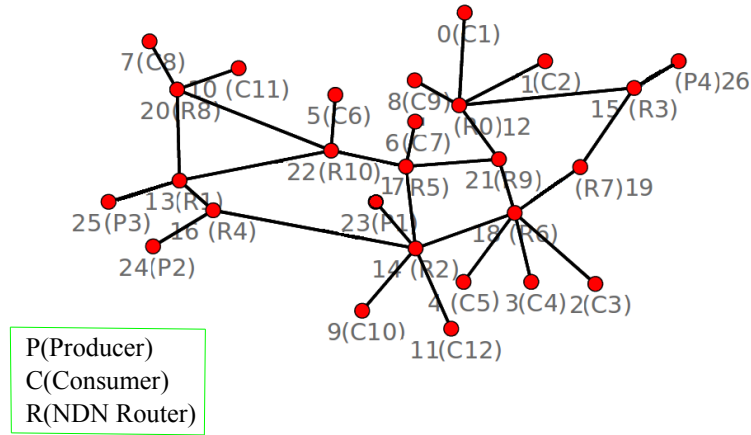


Figure 3. Topology of the Simulated Scenario With Abilene Core Routers.

As NDN attributes are inspired by the content-delivery and P2P networks, the Mandelbrot-Zipf distribution is known to model the consumer behaviors:

$$p(i) = \frac{\frac{1}{(i+q)^s}}{\sum_{j=1}^{|F|} \frac{1}{(j+q)^s}}, \quad i \in [1, |F|] \quad (13)$$

where $p(i)$ is the probability of retrieving the i th content from the $|F|$ contents in the network. The q and s parameters are fixed for an ISP. s is the Mzipf shaping in the range of $[0, 5]$. q is the plateau factor, controlling the plateau shape (flattened head) near the lowest ranked contents. If $q = 0$, then Mzipf degenerates into a Zipf distribution [37].

By applying this popularity distribution of files, it is observed that from 10^6 numbers of files in the catalog such as a large YouTube-like catalog, with $q = 5.0$ and $s = 0.5$, 811270 files, which is above 90% of user Interests, are forwarded towards their consumers [38]. If shaping factor, s , is increased to 2.0, then the number of most popular files in the catalog is decreased to just six files.

The experiments are repeated ten times, and all of them are implemented in 300 seconds of simulation time. A consumer node issues 20 requests per second. The evaluations are conducted in two different cases: enabled-cache consumers and disabled-cache consumers; in the former case, simulations are performed with the assumption of activated caches in consumers, and in the latter one, the caches of consumers are deactivated to examine the situation of having under pressure NDN routers. Each case consists of three types of simulation scenarios with three various requested content sizes: 1MB, 10MB, and 100MB. These various scenarios are selected in order to take a general decision based on the proposed probability caching strategy. In the first and second scenarios, ev-

ery consumer is running a single application produced by just one producer with the content size of 1MB and 10MB respectively; however, in the third scenario, the consumer is interested in each of the producers' 100MB-size contents.

In the first phase of Algorithm 2, the offline betweenness, online short-term HIs and online short-term PIs parameters are collected from NDN routers. Afterward LWMA of short-term parameters is calculated as accounted for in Section 2. Considering the numerical values of caching probabilities in Tables 3 and 4, related to enabled and disabled-cache cases respectively, the first scenario is carried out to compare the performance of the proposed method versus traditional betweenness probabilistic scheme [22]. As previously mentioned, ubiquitous caching strategy stores data chunks all over the path; however, it leads to a low network delay and is not an efficient use of router's cache [33]. Since the proposed method selects the router due to the combination of the long-term and short-term metrics, the probabilistic scheme is selected for comparisons in terms of average hit ratio of CS, the number of unique cache hits, and the round trip time of Interest-Data delay.

Table 2 represented the betweenness-centrality of NDN routers of Figure 3. Firstly, in order to make decisions for probabilities, betweenness values are normalized. Secondly, probability 0.5 is considered for a node if the numerical value of the allocated caching probability is lower than 0.5, otherwise, probability 1.0 is selected.

As observed in Table 2, the betweenness values of NDN-Router IDs 0, 2, 5, 6, 9, and 10 are high (*i.e.*, situated within a considerable number of paths between consumers and producers), leading to a caching probability 1.0 in comparison with other routers.



Table 1. System Parameters Used in the Experiments

	Parameter	Value
Network	MTU	1 KB
	Cache size	1GB
	Network size	27Nodes
	Topology	Abilene, with 11 Nodes
	Rate of Interest	20 Hertz
	Simulation time	300 seconds
	Number of seeds	10 (generated with normal distribution)
	Core network link capacity	10 Gb/s
	Access network link capacity	1 Gb/s
Catalog	Catalog size	10^6
	File size	1–100 MB
Popularity	Plateau factor in Mandelbrot-Zipf Distribution (q)	0, 5, 50
	Shaping factor in Mandelbrot-Zipf distribution (s)	[0.5, 3]
Forwarding & Caching	Forwarding strategy	BestRoute[34]
	Replacement policy	LRU

Table 2. Betweenness Probabilistic Scheme for the Simulated Scenario.

Parameter Router ID	Betweenness Centrality	Caching Probabilities	Final Caching Probabilities
NDN-Router 0	0.11	0.7	1
NDN-Router 1	0.05	0.2	0.5
NDN-Router 2	0.15	1	1
NDN-Router 3	0.04	0.2	0.5
NDN-Router 4	0.07	0.3	0.5
NDN-Router 5	0.14	0.9	1
NDN-Router 6	0.14	0.9	1
NDN-Router 7	0.03	0.2	0.5
NDN-Router 8	0.06	0.2	0.5
NDN-Router 9	0.13	0.8	1
NDN-Router 10	0.10	0.6	1

5.1 Enabled-Cache Consumer Case

In enabled-cache consumer case, the normalized metrics, and caching probabilities of the scenarios for the enabled-cache consumer case, computed by Algorithm 2, are shown in Table 3. In this table, there are three rows related to each NDN router showing the numerical values of the first, second and third scenarios respectively.

As remarked in Table 3, “LWMA of PIs” and “LWMA of Unique HIs” metrics gathered by the NDN-Router IDs 0, 2, 5, 6, 9 and 10 are as large as their betweenness values. It means that as these NDN

routers receive a large number of unique requests, they consequently collect a large number of incoming data chunks in their caches, by which more future Interests will be satisfied. Therefore, probability 1.0 is selected for these nodes. In NDN-Router ID 3, the numerous unique hits also ascribe a high probability of caching to this node. To maintain fairness, a probability equals to 0.5 is allocated to NDN-Router IDs 1, 4, 7, and 8, even though the output of several scenarios associated with these nodes display low numerical values.

In the first and second scenarios, every consumer is just running a single application, whereas in the third scenario, in order to extend the number of requested



Table 3. Normalized Parameters Gathered From the NDN Routers for the Enabled-Cache Consumer Scenarios.

Router ID \ Parameter	Betweenness Centrality	LWMA of PIs	LWMA of UniqHIs	Caching Probabilities	Final Caching Probability
NDN-Router 0	0.11	0.09091	0.09	0.5	1
	0.11	0.09091	0.08	0.4	
	0.11	0.09094	0.18	0.8	
NDN-Router 1	0.05	0.09091	0.06	0.2	0.5
	0.05	0.09091	0.004	0.2	
	0.05	0.09090	0.08	0.3	
NDN-Router 2	0.15	0.09091	0.16	1.0	1
	0.15	0.09091	0.38	1.0	
	0.15	0.09094	0.21	1.0	
NDN-Router 3	0.04	0.09091	0.10	0.3	1
	0.04	0.09091	0.29	0.5	
	0.04	0.09089	0.04	0.2	
NDN-Router 4	0.07	0.09091	0.09	0.3	0.5
	0.07	0.09091	0.09	0.4	
	0.07	0.09089	0.03	0.2	
NDN-Router 5	0.14	0.09091	0.06	0.5	1
	0.14	0.09091	0.004	0.5	
	0.14	0.09091	0.07	0.6	
NDN-Router 6	0.14	0.09091	0.09	0.6	1
	0.14	0.09091	0.08	0.5	
	0.14	0.09094	0.24	1.0	
NDN-Router 7	0.03	0.09091	0.06	0.2	0.5
	0.03	0.09091	0.004	0.2	
	0.03	0.09087	0.001	0.2	
NDN-Router 8	0.06	0.09091	0.09	0.3	0.5
	0.06	0.09091	0.004	0.2	
	0.06	0.09092	0.07	0.4	
NDN-Router 9	0.13	0.09091	0.13	0.7	1
	0.13	0.09091	0.008	0.4	
	0.13	0.09090	0.001	0.4	
NDN-Router 10	0.10	0.09091	0.06	0.3	1
	0.10	0.09091	0.004	0.3	
	0.10	0.09091	0.07	0.5	

applications, consumers are considered to be interested in all of the 100MB-size applications produced by the four producers. To simulate such a scenario, the shaping Zipf-Mandelbrot parameter is increased to $s = 3$ (*i.e.*, just a few files hold a higher popularity).

In Table 3, the values of LWMA of UniqHIs column are different in all the three scenarios, especially in NDN-Router IDs 0, 3, 6, and 9. Measurements show that in these nodes, CSs are operating differently. By performing the Randomized-SVD on the measurements of each scenario and normalizing the product,

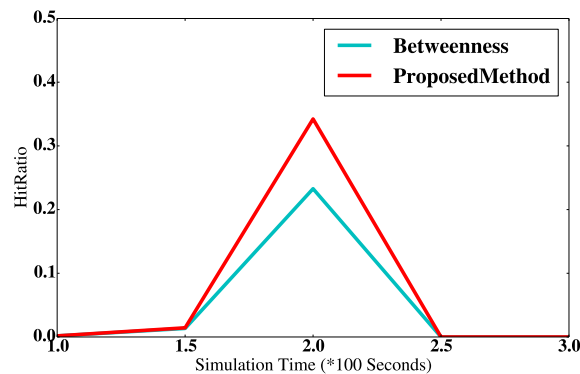


a value between $[0,1]$ is derived. Scikit-learn, an open source library for Python, was exploited for this purpose [39]. Thereafter, probability 0.5 is considered for the product if the numerical value of that scenario is lower than 0.5, otherwise, probability 1.0 is selected. Then the three rounded values of an NDN router are combined to determine the final caching probability and a logical OR operation is performed to combine the scenarios (last column of Table 3). The caching probability of 1.0 (caching everything) is allocated to an NDN router if any of these three rounded values are 1.0. In this way, by reducing the probability of the data-chunk caching placement in the regular nodes to 50%, the results of Table 3 are obtained. In other words, the nodes are clustered into two groups: important and regular groups. The important nodes have to cache every data chunk; however, the regular nodes just cache the received data chunks randomly with 50% probability.

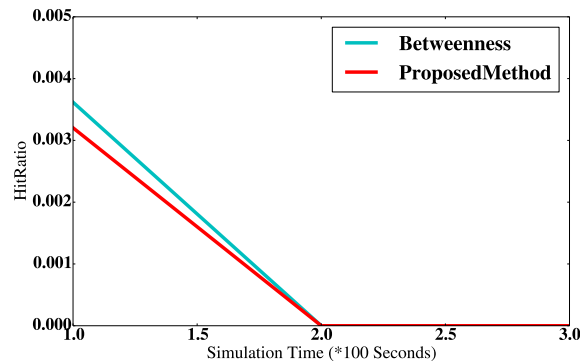
The performance comparisons between the proposed and betweenness methods are represented in Figure 4. Figure 4a exhibits that the proposed scheme outperforms betweenness scheme in terms of the NDN router's hit ratio. As observed, by assigning a probabilistic storage strategy to every CS, based on the proposed scheme, the average routers' hit ratio is increased to about 40%. It is consequently comprehended that, just by increasing the caching probability of NDN-Router ID 3 to one, the number of in-network satisfied requests of the proposed method is increased. Actually, by measuring the LWMA of PIs and Unique HIs, the proposed method can reveal the importance of NDN-Router ID 3 compared to the betweenness measurements. As a desired effect, the numbers of hit events are enhanced about 90% compared with the betweenness method. As a miss occurrence in NDN router's content store is normally due to the data not being stored in the cache, the unsatisfied request is forwarded to an upstream NDN router.

Therefore, at last one producer, in response to this unsatisfied Interest message, generates its requested content by the means of its CS module. As depicted in Figure 4b, the average hit ratio of producers' content stores in our scheme is 11.5% lower than the betweenness scheme. Accordingly, the management of in-network caching facilities in the proposed method helps to reduce the number of references to the producers' content stores, in turn resulting in decreasing producers' loads.

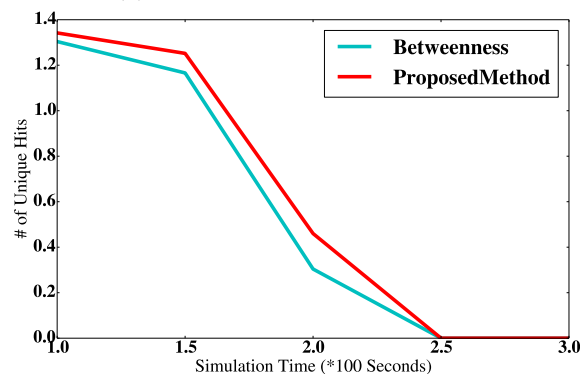
As portrayed in Figures 4a and 4b, until the middle of simulation runtime, whilst the NDN router's hit rate is ascending, the producer's hit ratio descends. Hence, the average hit rate of producers has a reverse relation with the average hit rate of NDN routers.



(a) Average Hit Ratio of NDN Routers.



(b) Average Hit Ratio of Producers.



(c) Average Number of Unique hit Interests in NDN Routers.

Figure 4. Betweenness and Proposed Methods With Content Size=1MB by Enabling Caches in Consumers.

To measure the importance of the NDN routers, one of the selected metrics is the number of unique hit Interests. In Figure 4c, the proposed method improves the average number of unique hit Interests compared to the betweenness-based strategy, since the proposed caching strategy obliged NDN-Router ID 3 to exploit its cache to store the passing data chunks.

As it can be noticed in Figure 4, requesting 1MB-size contents, fewer Interest packets are generated such that the consumer's requests are satisfied just by a few Data messages. However, as the size of the requested contents increases, the load on the NDN routers is also increased.



5.2 Disabled-cache Consumer case

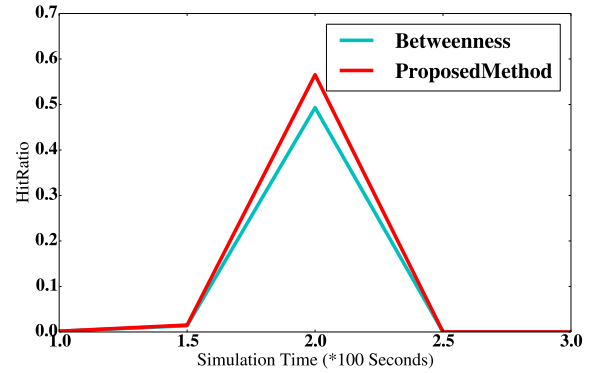
In the next simulations, it is shown that even by disabling the consumer's cache, the supremacy of the proposed NDN layer's caching scheme is still preserved. Therefore, in another set of simulations, we disabled the consumer's content store in order to lessen the cost of consumer-side devices and examine if it is probable to utilize the routers' caches in a more efficient manner. The scenarios are similar to the three ones running in the enabled-cache case, except that the consumer's caches are deactivated. The normalized metrics and caching probabilities of these scenarios, computed by Algorithm 2, are shown in Table 4. Similar to the previous case, NDN-Router IDs 0, 2, 3 and 6 accommodate more unique data chunks in their CSs compared to other routers. Therefore, the allocated caching probabilities and the hit rates are correspondingly higher.

As it was expected, in this case similar to the enabled-cache case, the number of PIs in PIT tables has not been changed for 1MB and 10MB content-size scenarios. It shows that as the content-size is increased to 100 MB, the number of Interest messages for the data chunks is also increased.

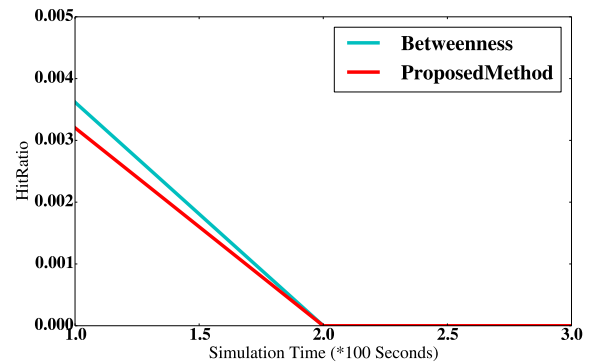
To compare the performance of the proposed method with the betweenness scheme, the first 1MB-content-size disabled-cache consumer's scenario, by considering the numerical values of the caching probabilities in Tables 2 and 4, is simulated again. Since the consumer's cache is disabled, the routers' caches try to compensate the loss of the disabled caches. It is observed that in the disabled-cache scenario, the hit rate of NDN layer is increased. As illustrated in Figure 5a and Figure 5c, the caches in NDN routers are satisfying the requests whose data chunks do not exist in consumers' caches. Furthermore, there is no need in referring to producer's CS.

Figure 5a summarizes that the proposed method achieves better overall hit ratio in NDN routers' caches compared to the betweenness scheme. In contrary to the enabled-cache scenario, the performance of the proposed scheme is nearly 15% more than the betweenness scheme.

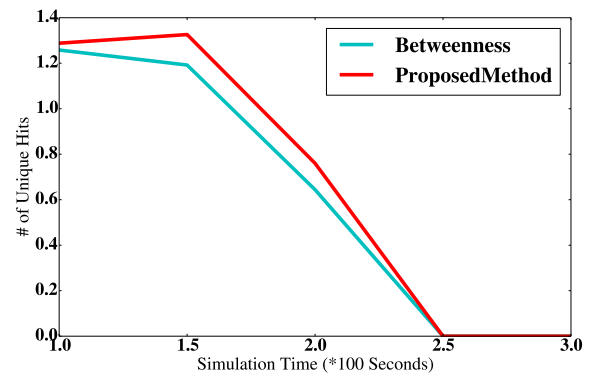
As Figure 5b demonstrates, comparing to the betweenness method, the proposed scheme decreases producers' hit ratio by approximately 11.5%. The main reason is that the high hit rate of an NDN router's cache reduces the reference rate to the corresponding producer's cache; *i.e.*, higher hit rate in one leads to lower hit rate in another. Compared with Figure 4b, the hit rates of producers' caches are approximately similar to the hit rates in enabled-cache case, showing that the NDN routers' caches do not compel the



(a) Average Hit Ratio of NDN Routers.



(b) Average Hit Ratio of Producers.



(c) Average Number of Unique hit Interests in NDN Routers.

Figure 5. Betweenness and Proposed Methods with Content Size=1MB by disabling Caches in Consumers.

producers to satisfy the Interests.

Figure 5c presents the average number of unique cache hits in NDN routers. In this disabled-cache case, the average number of unique cache hits in NDN routers is further than the enabled-cache case. In other words, by applying the proposed metric in selecting the important nodes, routers' CSs are utilized more than the enabled-cache case. In addition, the downward slope, compared to Figure 4c, is declining more gradually showing that CSs, in disabled-cache case, are being managed better than the enabled-cache case. As observed, until the first fifty seconds of simulation time, the proposed method, compared to betweenness



Table 4. Normalized Parameters Gathered From the NDN Routers for the Disabled-Cache Consumer Scenarios.

Router ID \ Parameter	Betweenness Centrality	LWMA of PIs	LWMA of UniqHIs	Caching Probabilities	Final Caching Probability
NDN-Router 0	0.11	0.09091	0.09	0.6	1
	0.11	0.09091	0.13	0.5	
	0.11	0.09098	0.25	1.0	
NDN-Router 1	0.05	0.09091	0.05	0.2	0.5
	0.05	0.09091	0.003	0.2	
	0.05	0.09083	0.0002	0.2	
NDN-Router 2	0.15	0.09091	0.14	1.0	1
	0.15	0.09091	0.31	1.0	
	0.15	0.09098	0.17	1.0	
NDN-Router 3	0.04	0.09091	0.16	0.8	1
	0.04	0.09091	0.19	0.4	
	0.04	0.09083	0.00001	0.2	
NDN-Router 4	0.07	0.09091	0.10	0.4	0.5
	0.07	0.09091	0.10	0.4	
	0.07	0.09083	0.0001	0.2	
NDN-Router 5	0.14	0.09091	0.06	1.0	1
	0.14	0.09091	0.03	0.5	
	0.14	0.09098	0.08	0.9	
NDN-Router 6	0.14	0.09091	0.09	1.0	1
	0.14	0.09091	0.14	0.6	
	0.14	0.09098	0.25	1.0	
NDN-Router 7	0.03	0.09091	0.051	0.2	0.5
	0.03	0.09091	0.003	0.2	
	0.03	0.09083	0.00002	0.2	
NDN-Router 8	0.06	0.09091	0.06	0.2	0.5
	0.06	0.09091	0.06	0.2	
	0.06	0.09098	0.07	0.4	
NDN-Router 9	0.13	0.09091	0.10	0.8	1
	0.13	0.09091	0.005	0.3	
	0.13	0.09083	0.000004	0.4	
NDN-Router 10	0.10	0.09091	0.06	0.4	1
	0.10	0.09091	0.03	0.3	
	0.10	0.09098	0.08	0.7	

method, enhances the reference rate to NDN routers' caches.

Figure 6 illustrates the RTT of the transmitted Interest and its delivered Data. The average delay of the proposed scheme, in both cases of disabled- and

enabled-cache consumers, decreases about 0.01% in comparison with the betweenness-centrality method. The reason is that, in both cases, the NDN routers help to satisfy Interest packets and reduce the request-response RTT from the user's standpoint. Here is the place where the caching role of NDN routers is



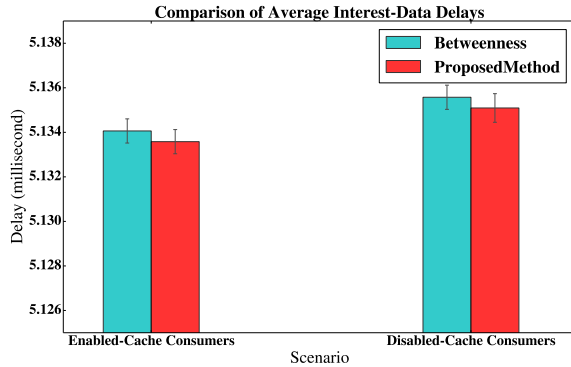


Figure 6. Average Round-Trip Time of Interest-Data Delay in Betweenness and Proposed Methods With Content Size=1MB by Enabling and Disabling Caches in Consumers.

ultimately proved worthwhile. In addition, it is illustrated that, in the enabled-cache scenario, by employing the caches in consumers, the RTT is less than the disabled-cache scenario. Therefore, by enabling consumers' caches, the request-satisfaction time can be improved.

6 Conclusion

The router's caching is one of the critical issues in NDN performance. In this article, the possibility of less in-network caching, by means of a new probabilistic caching strategy, is investigated. To assign a caching probability to each router in this network, a method is proposed based on both a long-term static metric (betweenness-centrality) and two short-term dynamic metrics (the number of Pending Interests and the number of unique satisfied/hit Interests) for an NDN router. After the short-term parameters are smoothed out by using LWMA, a data fusion technique based on Randomized-SVD is applied to extract the principal features of collected data set. Using these features, the caching probability of each router is obtained. Two typical scenarios based on Abilene core topology are implemented in NS-3 simulator to evaluate the performance of the proposed method.

The evaluation results could reveal that the proposed method does not just increase the average hit ratio of NDN routers, but it has also the advantage of exploiting information of graph-based centrality metric without considerable computation in nodes and message-passing between them. As assessed, in-network NDN routers have compensated the disabled caches of consumers. In addition, the caches of the overall NDN layer, in the enabled-cache consumer's case, perform better than betweenness scheme by finding the important routers to store data chunks.

Furthermore, it is observed that by allocating a proper probability for storing the data chunks and re-

ducing the unnecessary caching in NDN routers, it is possible to decrease the energy consumption while preserving a relatively good performance. Future efforts for extending this study involve applying mobility in NDN for real-time applications.

Acknowledgements

The first author would like to thank Dr. Neda Moghim, for the insightful comments and discussion in her defense meeting at the University of Isfahan.

References

- [1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2016-2021. Technical report, Cisco Systems, Inc., Feb. 2017.
- [2] Pekka Jäppinen, Renata Guarneri, and Luis M. Correia. An applications perspective into the Future Internet. *Journal of Network and Computer Applications*, 36(1):249 – 254, 2013. ISSN 1084-8045. doi:<https://doi.org/10.1016/j.jnca.2012.08.009>.
- [3] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard. Networking Named Content. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '09, pages 1–12, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-636-6. doi:[10.1145/1658939.1658941](https://doi.org/10.1145/1658939.1658941).
- [4] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, kc claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. Named Data Networking. *SIGCOMM Comput. Commun. Rev.*, 44(3):66–73, July 2014. ISSN 0146-4833. doi:[10.1145/2656877.2656887](https://doi.org/10.1145/2656877.2656887). URL <http://doi.acm.org/10.1145/2656877.2656887>.
- [5] D. Rossi and G. Rossini. On sizing CCN content stores by exploiting topological information. In *2012 Proceedings IEEE INFOCOM Workshops*, pages 280–285, March 2012. doi:[10.1109/INFCOMW.2012.6193506](https://doi.org/10.1109/INFCOMW.2012.6193506).
- [6] Yuemei Xu, Yang Li, Tao Lin, Zihou Wang, Guoqiang Zhang, Hui Tang, and Song Ci. An adaptive per-application storage management scheme based on manifold learning in information centric networks. *Future Generation Computer Systems*, 36(Supplement C):170 – 179, 2014. ISSN 0167-739X. doi:<https://doi.org/10.1016/j.future.2013.09.016>.
- [7] Narges Mehran and Naser Movahhedinia. Non-uniform EWMA-PCA based cache size allocation scheme in Named Data Networks. *Science China Information Sciences*, 61(1):012104, Aug 2017.



- ISSN 1869-1919. doi:10.1007/s11432-016-0501-5.
- [8] Amin Karami. ACCPndn: Adaptive Congestion Control Protocol in Named Data Networking by learning capacities using optimized Time-Lagged Feedforward Neural Network. *Journal of Network and Computer Applications*, 56 (Supplement C):1 – 18, 2015. ISSN 1084-8045. doi:https://doi.org/10.1016/j.jnca.2015.05.017.
- [9] Andrea Passarella. A survey on content-centric technologies for the current Internet: CDN and P2P solutions. *Computer Communications*, 35(1):1 – 32, 2012. ISSN 0140-3664. doi:https://doi.org/10.1016/j.comcom.2011.10.005.
- [10] Ibrahim Abdullahi, Suki Arif, and Suhaidi Hassan. Survey on caching approaches in Information Centric Networking. *Journal of Network and Computer Applications*, 56 (Supplement C):48 – 59, 2015. ISSN 1084-8045. doi:https://doi.org/10.1016/j.jnca.2015.06.011.
- [11] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos. A Survey of Information-Centric Networking Research. *IEEE Communications Surveys Tutorials*, 16 (2):1024–1049, Second 2014. ISSN 1553-877X. doi:10.1109/SURV.2013.070813.00063.
- [12] PARC's open-source implementation of content-centric networking, CCNx. <http://blogs.parc.com/ccnx/>. Accessed: November 2017.
- [13] Named Data Networking (NDN) A Future Internet Architecture. <http://named-data.net/>. Accessed: November 2017.
- [14] Michael Laszlo and Sumitra Mukherjee. A genetic algorithm that exchanges neighboring centers for k-means clustering. *Pattern Recognition Letters*, 28(16):2359 – 2366, 2007. ISSN 0167-8655. doi:https://doi.org/10.1016/j.patrec.2007.08.006.
- [15] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Matrix decompositions & latent semantic indexing in Introduction to Information Retrieval*, pages 220–235. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.
- [16] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *SIAM Review*, 53(2): 217–288, 2011. doi:10.1137/090771806.
- [17] Hervé Abdi and Lynne J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010. ISSN 1939-0068. doi:10.1002/wics.101.
- [18] Huamin Li, George C. Linderman, Arthur Szlam, Kelly P. Stanton, Yuval Kluger, and Mark Tygert. Algorithm 971: An implementation of a randomized algorithm for principal component analysis. *ACM Trans. Math. Softw.*, 43(3):28:1–28:14, January 2017. ISSN 0098-3500. doi:10.1145/3004053. URL <http://doi.acm.org/10.1145/3004053>.
- [19] W. Kaili, W. Muqing, Z. Min, and C. Yanqing. An autonomous system collaboration caching strategy based on content popularity in CCN. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6, Sept 2016. doi:10.1109/PIMRC.2016.7794940.
- [20] Ioannis Psaras, Wei Koong Chai, and George Pavlou. Probabilistic In-network Caching for Information-centric Networks. In *Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking*, ICN '12, pages 55–60, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1479-4. doi:10.1145/2342488.2342501.
- [21] S. Tarnoi, K. Suksomboon, W. Kumwilaisak, and Y. Ji. Performance of probabilistic caching and cache replacement policies for Content-Centric Networks. In *39th Annual IEEE Conference on Local Computer Networks*, pages 99–106, Sept 2014. doi:10.1109/LCN.2014.6925761.
- [22] Wei Koong Chai, Diliang He, Ioannis Psaras, and George Pavlou. Cache less for more in information-centric networks (extended version). *Computer Communications*, 36 (7):758 – 770, 2013. ISSN 0140-3664. doi:https://doi.org/10.1016/j.comcom.2013.01.007.
- [23] A. Araldo, D. Rossi, and F. Martignon. Cost-Aware Caching: Caching More (Costly Items) for Less (ISPs Operational Expenditures). *IEEE Transactions on Parallel and Distributed Systems*, 27(5):1316–1330, May 2016. ISSN 1045-9219. doi:10.1109/TPDS.2015.2433296.
- [24] Ulrik Brandes. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, 30(2):136 – 145, 2008. ISSN 0378-8733. doi:https://doi.org/10.1016/j.socnet.2007.11.001.
- [25] R. Winter, J. H. Schiller, N. Nikaein, and C. Bonnet. CrossTalk: cross-layer decision support based on global knowledge. *IEEE Communications Magazine*, 44(1):93–99, Jan 2006. ISSN 0163-6804. doi:10.1109/MCOM.2006.1580938.
- [26] Y. Zhong, Y. Fan, K. Huang, W. Tan, and J. Zhang. Time-Aware Service Recommendation for Mashup Creation. *IEEE Transactions on Services Computing*, 8(3):356–368, May 2015. ISSN 1939-1374. doi:10.1109/TSC.2014.2381496.
- [27] ndnSIM 2.0: A new version of the NDN simulator for NS-3. Technical Report NDN-0028, University of California, Los Angeles, January 2015. <https://named-data.net/wp-content/uploads/2013/07/ndn-0028-1-ndnsim-v2.pdf>.



- [28] R. Chiocchetti, D. Rossi, and G. Rossini. ccnSim: An highly scalable CCN simulator. In *2013 IEEE International Conference on Communications (ICC)*, pages 2309–2314, June 2013. doi:10.1109/ICC.2013.6654874.
- [29] Lorenzo Saino, Ioannis Psaras, and George Pavlou. Icarus: A Caching Simulator for Information Centric Networking (ICN). In *Proceedings of the 7th International ICST Conference on Simulation Tools and Techniques, SIMU-Tools '14*, pages 66–75, ICST, Brussels, Belgium, Belgium, 2014. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). ISBN 978-1-63190-007-5. doi:10.4108/icst.simutools.2014.254630.
- [30] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. A survey of information-centric networking. *IEEE Communications Magazine*, 50(7):26–36, July 2012. ISSN 0163-6804. doi:10.1109/MCOM.2012.6231276.
- [31] The Network Simulator-3 (ns-3). <http://www.nsnam.org/>. Accessed: November 2017.
- [32] Internet2 at Stanford. <https://itservices.stanford.edu/service/network/internet2>. Accessed: November 2017.
- [33] Behrouz Shahgholi Ghahfarokhi, Neda Moghim, and Shayan Eftekhari. Reducing channel zapping time in live TV broadcasting over content centric networks. *Multimedia Tools and Applications*, 76(22):23239–23271, Nov 2017. ISSN 1573-7721. doi:10.1007/s11042-016-4037-3.
- [34] Cheng Yi, Alexander Afanasyev, Ilya Moiseenko, Lan Wang, Beichuan Zhang, and Lixia Zhang. A case for stateful forwarding plane. *Computer Communications*, 36(7):779 – 791, 2013. ISSN 0140-3664. doi:https://doi.org/10.1016/j.comcom.2013.01.005.
- [35] Ioannis Psaras, Richard G. Clegg, Raul Landa, Wei Koong Chai, and George Pavlou. *Modelling and Evaluation of CCN-Caching Trees*, pages 78–91. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-20757-0. doi:10.1007/978-3-642-20757-0_7.
- [36] O. Saleh and M. Hefeeda. Modeling and Caching of Peer-to-Peer Traffic. In *Proceedings of the 2006 IEEE International Conference on Network Protocols*, pages 249–258, Nov 2006. doi:10.1109/ICNP.2006.320218.
- [37] Dario Rossi and Giuseppe Rossini. Caching performance of content centric networks under multi-path routing (and more). Technical report, Telecom ParisTech, 2011. <https://perso.telecom-paristech.fr/~drossi/paper/rossi11ccn-techrep1.pdf>.
- [38] Michele Tortelli, Luigi Alfredo Grieco, and Genaro Boggia. Performance assessment of routing strategies in Named Data Networking. In *Proceedings of GTTI 2013 Session on Telecommunication Network*, 2013.
- [39] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.



Narges Mehran received her B.Sc. degree in Computer Hardware Engineering (2012), and her M.Sc. in Computer Architecture (2016) from University of Isfahan, Isfahan, Iran. The title of her M.Sc. dissertation was “A Non-uniform Cache Size Allocation Scheme for Routers in Named Data Networks”. Her current research interests include Future Internet Architectures with an emphasis on Caching, Underwater Networking, and Cryptographic

Protocols on the Internet.



Naser Movahedinia received his B.Sc. from University of Tehran, Tehran, Iran in 1987, and his M.Sc. from Isfahan University of Technology, Isfahan, Iran in 1990 in Electrical and Communication Engineering. He got his Ph.D. degree from Carleton University, Ottawa, Canada in 1997. Currently he is a full professor at the Faculty of Computer Engineering, University of Isfahan. His research

interests are wireless networks, signal processing in communications and Internet Technology.

