WILEY | Hindawi

Research Article

Optimal Content Caching in Content-Centric Networks

Faiza Qazi,[1] **Osman Khalid,**[1] **Rao Naveed Bin Rais** ⓘ **,**[2]
Imran Ali Khan,[1] **and Atta ur Rehman Khan**[3]

[1]*Department of Computer Sciences, COMSATS University Islamabad, Pakistan*
[2]*College of Engineering and Information Technology, Ajman University, Ajman, UAE*
[3]*Faculty of Computing and Information Technology, Sohar University, Oman*

Correspondence should be addressed to Rao Naveed Bin Rais; r.rais@ajman.ac.ae

Content-Centric Networking (CCN) is a novel architecture that is shifting host-centric communication to a content-centric infrastructure. In recent years, in-network caching in CCNs has received significant attention from research community. To improve the cache hit ratio, most of the existing schemes store the content at maximum number of routers along the downloading path of content from source. While this helps in increased cache hits and reduction in delay and server load, the unnecessary caching significantly increases the network cost, bandwidth utilization, and storage consumption. To address the limitations in existing schemes, we propose an optimization based in-network caching policy, named as *opt-Cache*, which makes more efficient use of available cache resources, in order to reduce overall network utilization with reduced latency. Unlike existing schemes that mostly focus on a single factor to improve the cache performance, we intend to optimize the caching process by simultaneously considering various factors, e.g., content popularity, bandwidth, and latency, under a given set of constraints, e.g., available cache space, content availability, and careful eviction of existing contents in the cache. Our scheme determines optimized set of content to be cached at each node towards the edge based on content popularity and content distance from the content source. The contents that have less frequent requests have their popularity decreased with time. The optimal placement of contents across the CCN routers allows the overall reduction in bandwidth and latency. The proposed scheme is compared with the existing schemes and depicts better performance in terms of bandwidth consumption and latency while using less network resources.

1. Introduction

1.1. Background. With the emergence of new network applications and user requirements, there is a significant paradigm shift from traditional communication mechanisms to novel solutions. Tremendous growth in the number of users and user specific applications has evolved content-oriented and personalized information services to be the prevailing trend of network development [1]. The Internet has become largely a distribution network with massive amounts of video and web page content flowing from content providers to viewers. Internet users today demand more efficient, faster, and secure access to the data without being interested to know the physical location of content. The content-specific requirements

motivated the development of a content-aware infrastructure, which has been addressed to a certain extent through various solutions, such as CDN, P2P, and HTTP proxies, deployed on top of the current Internet [2]. However, the Internet architecture is traditionally host-centric and therefore cannot satisfy the growing content access requirements due to its lack of native support for the content-oriented data distribution service [1]. Such limitations brought a shift in network communication paradigm from location-based to content-based model.

Content-Centric Networking (CCN) is gradually evolving as an alternative approach to the conventional Internet architecture and is designed to be data-centric rather than the location-centric [1, 2]. The CCN rebuilds the Internet's

forwarding mechanism around the concept of content where nodes can issue *Interest* packets that are forwarded to content sources that can, in return, respond with the *data* packets. Each content is assigned a global identifier that all the network entities within an ICN can comprehend, which allows any router to cache data packets and, subsequently, serve future requests via its cache, rather than reforwarding the *Interest* to the origin of content. A classification of CCN and some important related challenges are presented in [3]. CCNs are designed to integrate with other application domains, e.g., edge computing, vehicular networks, Internet of Things (IoT), which can play supportive roles to augment CCNs in their functionality [4–6]. For instance, Li et al. proposed fog-enabled edge learning for cognitive CCN [4]. Fog computing-enabled content-centric 5G networks inherit the advantages of fog computing and content centricity [4]. The authors proposed an architecture that employs fog computing to elevate the cognitive ability of the CCN in order to realize context-awareness, timely response, and traffic offloading for 5G applications. In [5], the authors proposed a collaborative caching scheme to enable information-centric ability in IoT, which is beneficial for better connecting the devices and providing cooperative caching in fog computing. The authors in [6] proposed an architecture that combines Mobile Edge Computing with CCN in the context of connected vehicle environments, to achieve less response time for resource intensive services in fast changing networks. Figure 1 shows some of the areas where CCN can be deployed with existing supportive technologies.

1.2. CCN Caching. Ubiquitous in-network caching of content is a key feature of CCN that allows nodes on the request path to temporarily store a copy of the content requested from the original source [1]. This increases the content availability closer to the users, thereby reducing the response time of user's requests. However, in-network caching also encompasses various challenges including network topology complexity, limitations of cache placement and replacement policies, request routing, and caching traffic, to name a few. All these issues affect the performance of in-network caching [7, 8]. A content placement policy decides where and what content (or part of content) in the delivery path across the routers should be cached [1]. The default content placement policy caches a copy of content on each router along the downloading path [1, 9]. This avoids the need for routing *Interest* packets for the same content in future, thereby reducing delay and server load. However, the default caching policy is not resource-efficient and will cause huge caching redundancy cost due to storage of every content on each router, even if the same content is available at all neighboring nodes in the vicinity. For instance, it is reported in [9] that a CCN router with 10 TB cache space using Flash-based Solid-State Drives (SSDs) would cost $300,000 and consume 500 W of power.

1.3. Motivation. There are numerous research applications of content caching in wireless networks that includes, but not limited to (a) *content floating,* in which specific tagged

content is cached into some nodes confined within a specific geographic location [10], and the content is deleted from the nodes when they go out of the region; (b) *opportunistic data centers,* in which wireless nodes assembled in a group share their cache space to form only fly data centers to store content on the behalf of other nodes [11]; and (c) *mobile ad hoc network based caching,* in which the content is cached on mobile nodes in an ad hoc network based on consideration of multivalued attributes [12]. All the aforementioned applications of caching in the wireless environments lead to numerous challenges and open issues due to the uncertain conditions in wireless networks.

In the past few years, a number of caching schemes have been proposed in the literature, e.g., [5, 6, 9, 12, 13]. The coordination based schemes, such as [12, 13], share the network control information among nodes to coordinate for cache decisions. The coordination improves caching with diversity of contents and significantly reduces the redundancy, but with a tradeoff of control traffic overhead. The explicit cache coordination schemes, such as [14] assume that every router has prior knowledge of network topology, cache's state, and access frequency [13, 15]. These schemes reduce the cache redundancy to greater extent but introduce higher computational complexity. Moreover, explicit caching schemes do not take advantage of content knowledge, e.g., content popularity during caching. Alternatively, implicit caching schemes (e.g., [16]) require little control information exchange among nodes thus reducing the network overhead cost. Such schemes may either store content on a single router on the delivery path, or popular content may be redundantly stored on most nodes on the network [13, 15]. This may lead to underutilized spaces on the network or high content redundancy. To reduce the coordination cost, a few probabilistic schemes are proposed in which a node probabilistically makes individual decision whether or not to store the content. The scheme such as *ProbCache* [17] caches content probabilistically to reduce the redundancy, but with the tradeoff of cache miss. In contrast, the scheme, such as Leave Copy Everywhere (LCE) [2, 18] stores all the content to be delivered, and replace these content in the Least Recently Used (LRU) order. However, this approach consumes greater amount of buffer space, as each content is stored on every router along the path. The main objective of the scheme presented in [19] is to reduce cache redundancy using off-path central router, which attempts to improve the network performance in terms of content duplication and transmission delay. However, excessive bandwidth utilization for each caching update (cached, replaced, or dropped) is required by CCN routers to update the CCN Manager. Authors in [20] proposed a "Cache Strategy in Content Centric Networks based on Node's Importance." Most popular contents are placed at most important node in the network, where the importance of a node is calculated by its flow rate, which is based on number of users accessing node, request rate of content at node, and distance of the node from the rest of nodes in the network. However, the proposed scheme only focused on the geographical connectivity of the node and ignores content distribution in the network.
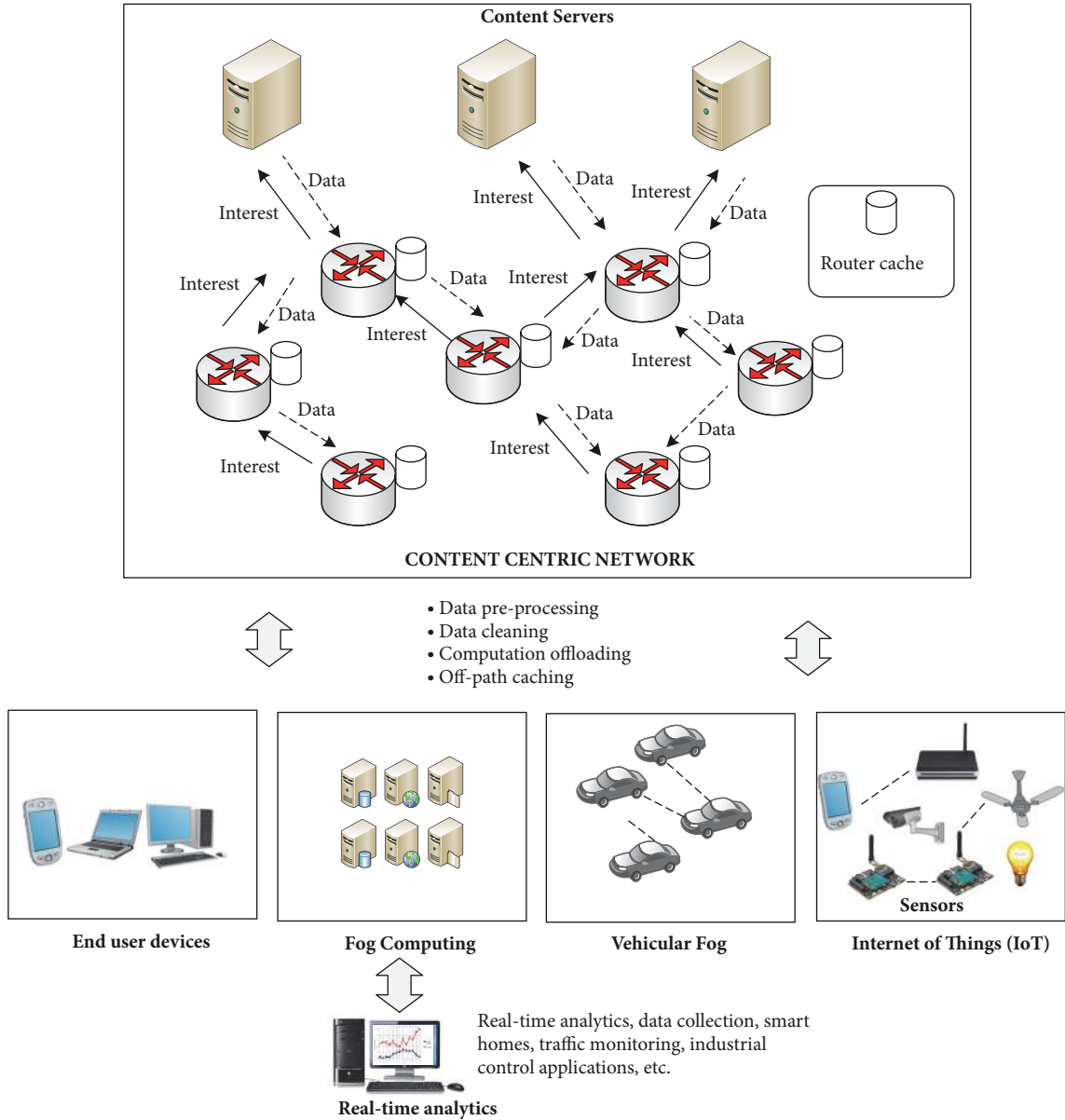
FIGURE 1: The supportive role of other communication technologies with CCN.

In all of the above discussed policies, we can find a tradeoff among content redundancy, bandwidth utilization, hop reduction, latency, and cache hits. Addressing one factor may affect the performance of other parameter(s). Although the content popularity and hop reduction in CCN have been discussed in some of the existing caching policies, such as [17, 19–26], the bandwidth consumption is still not optimal. Therefore, there is a need to devise an optimal caching strategy to increase the content availability while consuming minimum possible resources.

1.4. Contributions. The objective of this paper is to propose an optimization-based in-network caching policy, named as *opt-Cache*, which makes more efficient use of available cache resources, in order to reduce overall network utilization with reduced latency. Unlike existing schemes that mostly focus on a single factor to improve the cache performance, we intend to optimize the caching process by simultaneously considering various factors, e.g., content popularity, bandwidth, and latency, under a given set of constraints, e.g., available cache space, content availability, and careful disposal of existing contents in the cache. Our scheme determines optimized set of content to be cached at each node towards the edge based on content popularity and content distance from the content source. The proposed scheme aims to keep the content that are more popular for longer duration in the

cache and those content that have less frequent requests; their popularity is decreased with time. Such optimal placement of contents across the CCN routers allows the overall reduction in bandwidth and latency.

The following are our main contributions of this paper:

(i) We propose an optimized caching policy, named as *opt-Cache* that reduces the content retrieval latency by caching content near the user at the edge.

(ii) We design an in-network caching mechanism that jointly considers content popularity and hop reduction to reduce bandwidth consumption.

(iii) The *opt-Cache* caches popular content for longer time and therefore decreases the cache operation.

(iv) To demonstrate the effectiveness of *opt-Cache,* we perform simulation in Icarus [18], an ICN simulator specifically designed for analyzing caching and routing polices in ICN. We compare *opt-Cache* against existing caching polices using GEANT [18] a real-world internet topology. We study the impact of various parameters such as cache size, content popularity, and stretch on our policy. We find a significant improvement in latency, cache hit ratio, stretch, and link load as compared to the existing schemes.

The rest of the paper is organized as follows. In Section 2, we present the most relevant work. The system model is presented in Section 3. In Section 4 we present the results and discussions, and finally Section 5 presents conclusion and future work.

2. Related Work

Network caching policies in CCN have attracted the attention of many researchers in the recent years. Although parameters such as content popularity and hop reduction in content caching have been dealt in some existing policies, such as [8, 18, 27, 28], yet bandwidth consumption still needs some attention. Caching scheme, such as Leave Copy Down (LCD), copies the content at the direct neighbor of requesting node after a cache hit occurs [18, 29]. The LCD algorithm is aimed at keeping content as closer to user as possible. Psaras et al. [17] focus on the distribution of content in the routers' cache using the in-network caching concept. The authors propose a *ProbCache* scheme that caches content probabilistically to efficiently manage cache resources by reducing cache redundancy. The scheme allows caching space to other traffic sharing the same path. The *ProbCache* prefers keeping large cache at edge. However, content popularity distribution is not considered by the scheme. Moreover, the *ProbCache* incurs high computational cost.

Xu et al. proposed an energy efficient information-centric multimedia streaming framework for vehicular networks (GrIMS) [30]. The authors adopted a multiple windows' queuing model with differentiated service rates to characterize the multimedia retrieval in heterogeneous wireless vehicular ad hoc networks (VANETs). First, the authors model a cost optimization model to address the tradeoffs between energy efficiency and quality of experience (QoE). However, in real-world scenarios solving such models becomes practically infeasible due to time varying conditions, e.g., mobility of vehicles. Therefore, the authors propose a set of heuristic algorithms to achieve both high QoE levels and energy saving in GrIMS's cloud-based processing, multipath selection, and in-network caching context.

In [31], the authors proposed an ICN based cooperative caching solution for multimedia delivery in VANETs. The paper focuses on exploring the potential social cooperation among neighbor vehicles in highway to improve the utilization of caching resources. The authors proposed a highway traffic model with multiple lanes and considered two types of vehicles in context of social based cooperative caching, the partners and couriers, jointly based on location and lane information. The proposed model aims at maximizing video playback quality without excessive startup delay and minimizing the playback freezing ratio. Quan et al. proposed a CCN-customized lookup engine that utilizes Tree-Bitmap and Bloom-Filter mechanisms to significantly reduce the lookup time [32]. The proposed approach splits the CCN prefix into a constant size T-segment and a variable length B-segment with a relative short length, which are treated using TB and BF, respectively. In a similar proposal, Quan et al. proposed a name lookup solution with adaptive prefix bloom filter (NLAPB) in which each NDN name/prefix is split into B-prefix followed by T-suffix [33]. B-prefix is matched by Bloom filters whereas T-suffix is processed by the small-scale trie. The length of B-prefixes (and T-suffixes) is dynamically adjusted based on their popularity in order to accelerate the lookup. The proposed approach reduces the lookup time significantly.

In [34], the authors consider the problem of content delivery in fog architecture with time varying content popularity. The authors perform an information theoretic analysis to optimize front-haul link capacity, cache storage, and wireless bandwidth. A high-SNR latency metric is adopted to capture the performance of online cache design. The serial transmission mode is considered initially where edge nodes start transmission on the wireless channel after completion of cloud to edge node transmission. Then, pipeline delivery mode is investigated that allows simultaneous transmission and reception of information by edge nodes. The authors concluded that the rate of front-haul link sets fundamental limit on the achievable latency performance. The authors in [35] perform information-theoretic formulation of caching problem that utilizes coded caching approach to achieve global caching gain, in addition to local caching gain. The gain depends on aggregate global cache size, and is derived from jointly optimizing the content placement and delivery phases, so as to satisfy different delivery demands with a single multicast coded transmission.

Femminella et al. proposed an architecture based on distributed caching for the distribution of dynamic sensor data [36]. The authors utilized NetServ hosting environment to perform service modularization and virtualization of network nodes, implemented with an aim of extending the functions of network nodes. The information collected from a number of distributed sensors is made available to a large

number of recipients/software components at almost real-time, e.g., augmented reality servers, security engines, fault recovery applications, etc. The simulation results indicate a downward trend in latency and overall network traffic. In [37], the authors discuss various insights of caching or replicating genome data set to provide guidelines for the improvement and optimization of genome content distribution. The authors analyze the main genome processing applications, and classify genome workflow computation alternatives as either a location machine or a distributed cloud. The authors also discuss the main networking topologies available to efficiently support the genome as a service (GaaS) solution. The authors highlight the challenges accompanied with the temporal evolution of genome content popularity, and suggest that special investigations for managing data storage could significantly contribute to the optimization of genomic content distribution. The authors argue that discovery of hidden relationships between different data sets could allow anticipating popularity variations in of some of them, thus facilitating the management of geographically distributed replicas. Moreover, the security and privacy issues in genome data caching are also touched. Due to existence of personal information in genome data, the authors conclude that caching or replication of genomics datasets seems feasible only for anonymized contents for which informed consent has been issued.

In [19], the authors propose cooperative caching policy where on-path caching is facilitated by off-path routers that are strategically placed by service provider. However, the bandwidth utilization of the proposed scheme is very high, because all the edge routers have to send their state information to ICN manager. Works proposed by [20, 21] investigate that caching only at a subset of node(s) along the content delivery path can improve in-network caching performance in terms of cache and server hit. To overcome the limitation of LCE, a native caching policy *Cache Less for More* (CL4M) is proposed by Wei Koong Chai et al. [21]. The proposed scheme caches at only one chosen intermediate node for each request along the delivery path. The concept of betweenness centrality is used, which is a measure of number of times a specific node resides on the shortest delivery path for all pairs of nodes in a network topology. Caching at such node will not only minimize cache replacement but also increase cache hit ratio by caching where cache hit is most probable to happen. Latency-aware caching (LAC) is proposed in [22] to reduce average latency to retrieve content from any router. It determines a set of content to be cached at each node in the network [22]. Based on request rate of a content from users, the scheme starts caching the most popular content at network edge. After caching content at edge, the algorithm recalculates the relative popularity of the content based on request miss from downstream to cache content in core.

Ren et al. [23] propose a distributed caching scheme of ICNs that reduces bandwidth consumption and limit cache operations by considering input/output operations of cache storage. APC [38] supports energy attempts to target efficient content distribution by reducing the hop count and caching only the frequently requested content by considering the

available cache space. However, the scheme is compared only with LCE and lacks the theoretical details and supporting simulation results. In [39], the *CachinMobile* is proposed in which the energy consumption is minimized by using Device to Device (D2D) communication. Various potential issues related to dynamic mobility, bandwidth resource scheduling and interference management are ignored. The research in [40] utilizes user mobility and trajectories' information for proactive and adaptive caching at Base Stations and User Terminals. Future position of the user can provide seamless handover and content download for user in case of proactive caching. However, as user terminals are growing enormously, collecting data for all the users for adaptive caching can be challenging. Moreover, motivating the users to use D2D communication is challenging.

The authors in [41] argue that it is not necessary for a node to have a higher graph connectivity to cache content optimally. To find a node that can cache large number of content, they propose a content-based centrality metric that takes into account how well a node is connected to the network content. The most frequently accessed content are placed at a node with highest Content Base Centrality (CBC), where CBC of a node N is defined as the ratio between the numbers of shortest paths among all users through which the content traverse via node N, to the total number of shortest paths through which content traverse among users. However, the proposed scheme only focuses on content distribution and ignores geographical location of the node. In [13], the authors have proposed a Multiattribute Caching (MAC) strategy that computes the rank of a content by computing the weighted sum of multiple attributes, such as hop count/distance, content popularity, degree of node, and available cache space. The weights are set to be adaptive and are computed accordingly to the overall network condition. In another proposal, the same authors proposed a caching strategy for mobile ad hoc networks [12]. The optimal node for content placement is selected using the weighted approached presented in [13, 15]. The weights are made adaptive to network conditions using backpropagation method. The new caching node is selected if the previous node moves to a disconnected region. However, in both the aforementioned schemes [12, 13], optimization is not used.

The authors in [24, 25] propose content caching schemes for Vehicular Ad hoc Networks (VANET) using ICNs. In [24], the main objective is to reduce cache redundancy and lower caching replacement overhead. In [25], the authors propose an ICN based cost-efficient multimedia delivery model for VANET. However, the data collection due to high mobility in dynamic topology is challenging in these strategies. Moreover, finding an energy efficient route between source and destination is also challenging. In [26], the authors investigate the challenging issue of scalable video streaming in ICN. The proposed scheme improves the user's QoE by providing requested content quickly. However, the scheme lacks the parameter, such as received video quality.

In all of the above policies, limited attention has been given on cache frequency reduction and hence saving

of bandwidth and minimizing content' retrieval latency are ignored. The objective of our proposed scheme is to keep the popular content in the cache for longer time to increase hit ratio and reduce server access. In the next section, we present an overview of our proposed *opt-Cache* scheme.

3. System Model

In this section, we present the overall architecture of the proposed scheme along with its major components description. We also provide the message types supported by the proposed scheme.

3.1. Major Components. We model a CCN network as a connected graph $G = (V, E)$, consisting of $V = \{V_1, V_2, V_3, \ldots, V_n\}$ nodes and $E = \{e_1, e_2, e_3, \ldots, e_m\}$ links. Let $F = \{f_1, f_2, f_3, \ldots, f_k\}$ be the set of available content and $S = \{S_1, S_2, S_3, \ldots, S_r\}$ be the set of servers in the network. Initially, all contents are assumed to be available only at the network servers, which are directly connected to routers. Following are the major components of the proposed *opt-Cache* scheme.

(i) End Nodes/Users. A CCN user is the request originator. It initiates the request and forwards to the connected edge routers. A user sends an *Interest* packet for the interested content over an available connection. Any party receiving the *Interest* packet and having data that matches the *Interest* packet may transmit matching content using Data packet to the user.

(ii) CCN Routers. CCN routers are standard routers with added caching capabilities. Each CCN router is provided with extra storage space to cache content passing through it, which enables it to serve the subsequent requests for the same content from its local cache. Delivery of requested content must be ensured either by CCN router or source server. In our proposed caching method, if the cache of the CCN router is full, then the Least Recently Used (LRU) cache policy is incorporated to cache the newly arrived content. Each node has three data structures: Pending Interest Table (PIT), Forwarding Interest Base (FIB), and Content Store (CS) [20] as shown in Figure 2.

(iii) End Nodes/Source. With respect to CCN content, there are two types of sources, CCN routers and CCN servers. CCN servers are the content originators or permanent source of content whereas CCN-Fog routers store content temporarily in local cache.

3.2. Assumptions. For simplicity' we assume that the cached content have same size [17, 22]. To get the popularity of the requested content, each intermediate router counts the frequency of the requested content [18]. Each caching node can obtain hop count, i.e., the number of hops from it to the original server with the help of TTL associated with data packet. For the rest of the paper' we use word "router" or

"cache node," interchangeably. Table 1 shows the definitions of the notations used in the paper.

3.3. Optimization Framework. The objective of the proposed *opt-Cache* scheme is to reduce the resource consumption in obtaining the requested content. We address the problem of how a router with limited caching capacity caches content in the network to minimize latency and bandwidth utilization. Therefore, we model the objective function as a minimization problem for optimal content placement, as indicated in

$$\min \sum_{v \in V} \sum_{f \in F} \lambda(f, v) \times \sum_{u \in V} T(f, u, v) \cdot Y(f, v, u), \quad \text{(1a)}$$

s.t.

$$\sum_{f \in F} X(u, f) \times size(f) \leq C_i(v), \quad \forall v \in V \quad \text{(1b)}$$

$$Y(f, u, v) \leq X(u, f), \quad \forall u, v, f \quad \text{(1c)}$$

$$\sum_{u \in V} Y(f, u, v) = 1, \quad \forall v, k \quad \text{(1d)}$$

$$X(f, v) = \{0, 1\},$$

$$Y(f, u, v) = \{0, 1\}, \quad \text{(1e)}$$

$$\forall f, u, v$$

In the network, any node v may request the content f with a rate λ that may take T time units in transferring from source to the requesting node. Whether or not the particular content f is downloaded for caching depends on the value of binary decision variable Y. With this caching policy, (1a) represents our objective function to minimize overall distance of contents from the requesting nodes, thereby reducing the latency, (1b) describes the constraint for finite cache size at each router, (1c) indicates that a node can download the copy of a content to cache, only if the content is available in the source node, (1d) signify that each request must be entertained by at least one router, and finally (1e) indicates the possible values of decision variables.

In (1a), (1b), (1c), (1d), and (1e), we want to minimize the overall distance of content from the requesting node, by making use of request rate and time. In simple words, the product of *request rate for a content* and *time consumed by a node to request content from source* gives the distance measure, which we want to minimize. Since we want to minimize the overall distance, thus latency, therefore, the content will be downloaded from the source and will be cached on a node that will reduce overall latency. To achieve our objective of minimizing latency for all requests, we need to decide on whether to cache the requested content or not, subject to the large number of content in the network and finite cache capacity of the node. With a limited cache size, each router while serving an Interest packet intelligently decides whether or not to cache the requested content and what the best possible location to cache the content is. Basically,
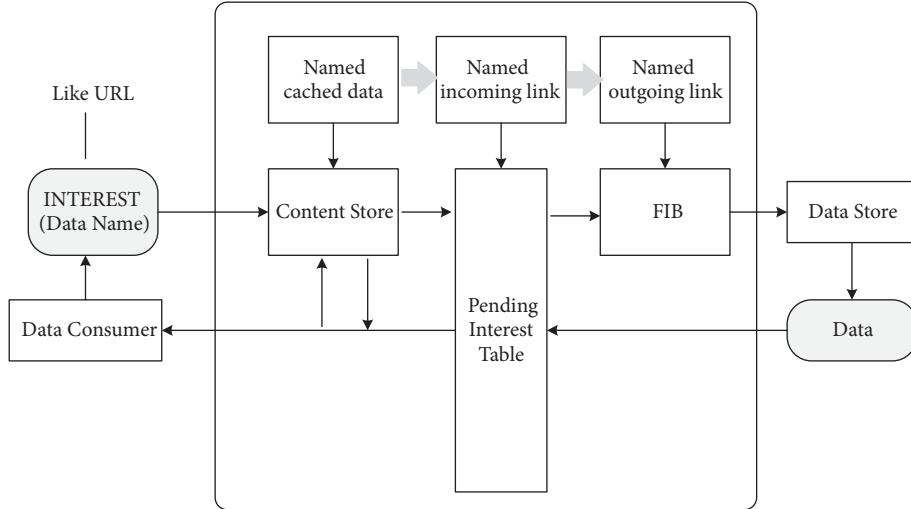
FIGURE 2: Data flow with-in CCN.

TABLE 1: Notations and their meanings.

Parameters	Meaning
V	Set of routers in the network
F	Set of content items
F_N	Set of content items in a node N
$\lambda(f, v)$	Request rate of $f \in F$ at $v \in V$
$C_i(v)$	Available cache size of node $v \in V$
$Size(f)$	Size of the content $f \in F$
$Src(f)$	Original source for $f \in F$
$D(u, v)$	Distance between request and source node in number of hop count
R_h	Number of nodes the request traversed from requestor to source
C_h	Number of hops, the content traversed during content delivery
$T(f, u, v)$	Time consumed by node u to download content f from node v
	Decision Variables
$X(u, f)$	Takes value 1 if node u caches copy of content f_k and 0 otherwise
$Y(f, v, u)$	Takes value 1 if node u downloads a copy of content f_k from v, and 0 otherwise

problems (1a), (1b), (1c), (1d), and (1e) define our objective function that we want to optimize, i.e., to reduce the overall cache latency time, by placing the desired contents near to the requesting nodes. However, it is difficult to solve the afore-stated optimization problem in a centralized and globally valid way because the variables that feed the problem are time-varying. The randomness in the requests and the dynamically changing available cache sizes, content sizes, and bandwidth makes it difficult to solve the optimization problem as once if the solution is found, the conditions are already changed and the solution cannot be applied to

the system. Therefore, a heuristic algorithm (Algorithm 1) is utilized to achieve the nearest optimal solution that makes use (2), which represents the objective of problems (1a), (1b), (1c), (1d), and (1e), i.e., to place the frequently requested content at the minimum distance from the requesting node and thus to reduce the overall latency.

To achieve that, each CCN router may store frequently-requested content by users to increase probability of hit. Further to minimize transmission delay, each router may cache content from a distant source. Based on above mechanism, each on-path router computes Cache Gain Probability

```
(01)   Available cache size=Ci;
(02)   For a request for content f (k), do the following
(03)       Update maximum rate of request: λ(f, v)_max
(04)       Update maximum possible hops D(v, u)_max
(05)       if cache hit on router then
(06)           CGP (f) = λ(f,v)/λ(f,v)_max × D(v,u)/D(v,u)_max × C_h/R_h.
(07)       else
(08)           CGP (f) = 1/λ(f,v)_max × D(v,u)/D(v,u)_max × C_h/R_h.
(09)           CGP(f) = CGP(f) + (1 − CGP(f)) × P_init      // Update CGP of the accessed content
(10)           CGP(j) = CGP(j) + γ^k                         // Update the CGP of rest of contents
(11)       end if
(12)       if Ci > 0 then
(13)           cache content f (k)
(14)       else if Ci == 0 then
(15)           if CGP(f) ≥ min{CGP(j)}}, ∀j ∈ F_N then
(16)               cache content f
(17)           end if
(18)       end if
(19)   end if
```

ALGORITHM 1: On-path content caching on CCN router.

(CGP) of content f for each caching content using following formula and compares it with the score of existing cached content:

$$CGP\ (f) = \frac{\lambda\ (f, v)}{\lambda\ (f, v)_{max}} \times \frac{D\ (v, u)}{D\ (v, u)_{max}} \times \frac{C_h}{R_h}. \qquad (2)$$

where $\lambda(f, v)$ is the request rate of content f at node v and $D(v, u)$ is the distance traversed by the content to reach node v from node u, i.e., the number of hop counts traversed. The parameter $D(v, u)$ can be obtained with the help of TTL associated with data packet. $\lambda(f, v)$ is normalized request rate of content (normalized with maximum request rate $\lambda(f, v)_{max}$ of cached content). $\lambda(f, v)_{max}$ is the maximum rate of requests at router at any instance of time. It is the maximum frequency of a request recorded by a router for a content f, and $D(v, u)_{max}$ is maximum hop count traversed by the content to reach node v from node u. If the cached content is served through the node adjacent to the requesting node, then $D(v, u) = 1$ and $\lambda_{new}(f, v) = \lambda(f, v)/\lambda(f, v)_{max} + 1/\lambda(f, v)_{max}$. If a new content (not cached) is served, then $\lambda(f, v) = 1/\lambda(f, v)_{max}$ and $D(v, u)$ is normalized hop count for this node. In case a content is cached at CCN router, $\lambda(f, v)$ increases linearly for every new request for the same content. If a content that is not cached at CCN router is served, then its $\lambda(f, v) = 1$. Equation (2) aims to assign the priority to content based on their popularity and distance from the original source, thereby, caching a more popular content, with higher request rate, from a distant source to the nearest possible nodes. Such mechanism will in fact reduce the distance and thus latency as defined in (1a), (1b), (1c), (1d), and (1e). As the content is placed near the requesting node, the cache gain probability of the

content increases, as the number of hops traversed will be decreased.

Furthermore, every time a cache hit occurs, we update the CGP of f so that content that are frequently accessed have high CGP. The following is used to update CGP, where $P_{init} \in (0, 1)$ is initialization constant:

$$CGP\ (f) = CGP\ (f) + (1 − CGP\ (f)) \times P_{init}. \qquad (3)$$

Similarly' CGP of the other contents that are not accessed must be aged with their value being reduced in the process. In the following, $\gamma \in (0, 1)$ is the aging constant and k is the number of time units elapsed since the last time the metric was aged:

$$CGP\ (j) = CGP\ (j) + \gamma^k. \qquad (4)$$

The proposed heuristic method for on-path caching is presented in the Algorithm 1 which runs on a CCN router on receiving request.

3.4. Complexity Analysis of Algorithm 1. Algorithm 1 follows a simple linear time complexity class. Each statement is iterated once from Line (2) to Line (14); therefore, the complexity is $O(1)$. At Line (15), the contents stored on a node are iterated once, with a complexity of $O(|F_N|)$. This makes the overall complexity of Algorithm 1 of the order $O(|F_N|)$.

4. Results and Discussions

In this section, we describe the experimental setup and simulation results. We compare the performance of proposed

TABLE 2: Simulation setup.

Parameter	Value
No. of warm-up requests	40,000
No. of measured requests	40,000
Popularity model	0.6, 0.8, 1.0
No. of content universe	10,000
Cache size	4% – 20%
Request rate	1.0 request/sec

scheme *opt-Cache* against existing caching policies, LCE, LCD, ProbCache, and CL4M (illustrated in Section 2 related work). The aforementioned schemes are selected being relevant and exhibiting consistent performance, hence most cited in the literature.

To route an Interest packet from client to source, we assume the network uses Dijkstra's shortest weighted path routing, where delay on the links corresponds to weights. If the content is found at any intermediate router towards source, then it is served from that router cache. To demonstrate the effectiveness of *opt-Cache* policy, we perform simulation in Icarus [18], a CCN simulator specifically designed for analyzing caching and routing polices in ICN. Scenario generation, experiment orchestration, experiment execution, and results collection are the four basic building blocks of Icarus. In simulation, each content request is considered as an event. Therefore, whenever an event occurs the corresponding timestamp, receiver, and source of event are stored. The result collection block of the simulator gathers the result of simulation. Through the sum of delays on each link traversed during content download, the latency is calculated in Icarus.

Extensive experiments are performed on real-world topology GEANT, which consists of 40 nodes and 60 edges, is an academic network spread around the world, and results for 95% confidence intervals are plotted. Table 2 describes our simulation set up.

In our simulation, the caches are initialized with first 10,000 content and subsequent requests are used for performance evaluation. Zipfian distribution with skewness $\alpha \in [0\ 1]$ is assumed as probability of requesting a content. Originally content are stored and uniformly distributed in content sources (server). Nodes with degree one are considered as users/clients. The number of contents (content universe) is set as F=1000. The network cache size varies from 4% to 20% of total content universe. Content popularity skewness (α) varies from 0.6 to 1.0, where α with value 0.6, 0.8, and 1.0 refers to low popularity model, normal popularity model, and high popularity model, respectively.

To compute the efficiency of our optimized content cache strategy' we use different performance metrics, such as latency, cache hit ratio, and path stretch. These performance metrics can be calculated as follows:

(i) Latency. Latency is number of time units (in millisecond) a content take to reach at user or the delay taken to deliver a content [23] and can be calculated as follows:

$$Latency = request\ travel\ delay \\ + content\ travel\ delay. \tag{5}$$

(ii) Link Load. Link load is the number of bytes a link traversed per unit time to retrieve the requested content and can be calculated using [23]

$$link\_load \\ = \frac{request_{size} \times request_{link\_count} + content_{size} \times content_{link\_count}}{duration}, \tag{6}$$

where

$$duration = content\ retrievel\ time \\ - content\ request\ time, \tag{7}$$

$request_{size}$ is the size of request in bytes, $request_{link\_count}$ is the number of links traversed by the request to reach the source, $content_{size}$ is the size of the content to retrieve, and $content_{link\_count}$ is the number of links traversed by the content to reach the request originator.

(iii) Hit Ratio. For a request, if the content is found in a node's cache, then we call it a "hit"; otherwise, it is known as a "miss." When cache miss operation occurred, the contents are retrieved from server. Cache hit ratio, i.e., the portion of requested content served by node cache, measures the efficiency of routers [18] and can be calculated as

$$Hit\_Ratio = \frac{cache_{hits}}{cache_{hits} + servers_{hits}}. \tag{8}$$

(iv) Path Stretch. We consider the path stretch d/|P| as the number of CCN backbone hops d that the data chunk has actually traveled in the network, normalized over the path length |P| until the content originator (i.e., without caching).

4.1. Discussion on Latency Performance. In this section, we discuss the latency performance of our *opt-Cache* strategy for a variety of different settings.

4.1.1. Latency Results for Different Caching Strategies at Different Popularity Rates. Figures 3(a), 3(b), and 3(c) show the latency results for caching strategies at different popularity rates ranging from 0.6 to 1.0. From derived results we observe that *opt-Cache* outperforms the existing policies by 4%–18% for a wide range of cache sizes (ranges from 4%–20% of content universe). Better cache utilization is the primary reason behind the superior performance of optimized cache strategy.
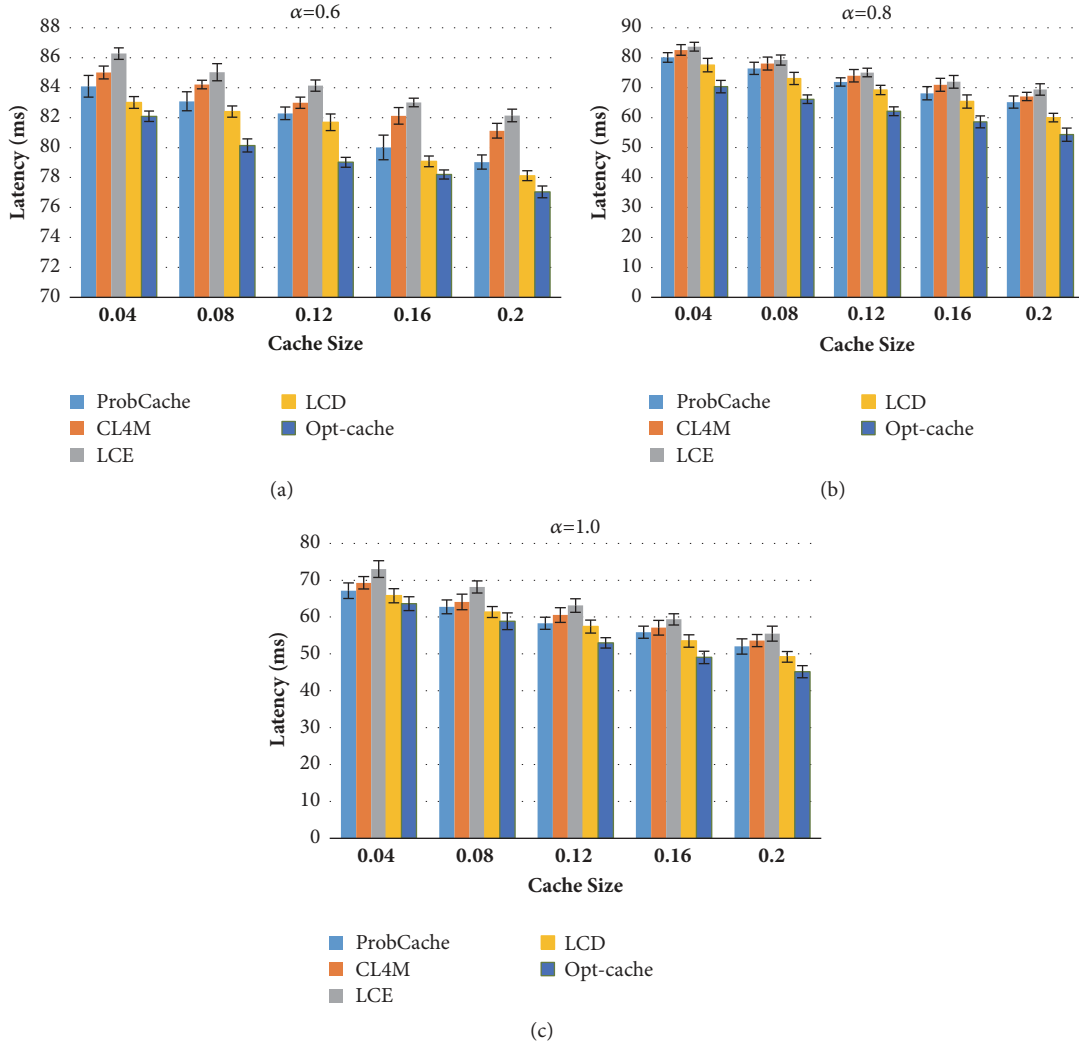
Figure 3: Latency performance with varying cache size and popularity parameter α.

4.1.2. Latency Performance Using Different Content Universe. Figures 4(a), 4(b), and 4(c) reflect the performance of *opt-Cache* for GEANT when using different content universe. Results are generated on fixed cache size, which is about 5% of content size. Content universe varied from 10,000 to 50,000. Our simulation results show that performance of *opt-Cache* is superior to other strategies for large content universe as well. As the content-to-cache size is kept constant; therefore, with increase in content universe, absolute cache size increases causing latency decrement and cache utilization increases as popular content are readily available in cache. Figure 4 also depicts the scalability of *opt-Cache*. These results show that our algorithm can work with large content universe with consistent performance.

4.2. Discussion on Cache Hit Ratio. Hit ratio is an important performance parameter for CCN caching strategies. Cache hit occurs when a request is fulfilled by cache router, while a miss encounters if request can only be served by original content source. High hit ratio demonstrates that more requests are being served by intermediate cache routers, thereby reducing load on original source. We compare the cache hit ratio of *opt-Cache* with existing strategies, ranging from low to high popularity rate models (α=0.6, α=0.8, and α=1.0). We perform simulations on different cache sizes. Figures 5(a), 5(b), and 5(c) show the cache hit ratio of *opt-Cache* at different popularity models with different cache sizes is superior to the existing techniques. The proposed scheme shows an improved performance in terms of hit ratio compared to the existing. The primary reason is that a router caches the content only if an optimal solution is found for transfer time from the source and request rate. We observe that the hit ratio increases with increase in ratio of cache size to content, and the content popularity. The bigger the cache size ratio, the greater the cache hit ratio; however, *opt-Cache* outperforms the rest of the strategies
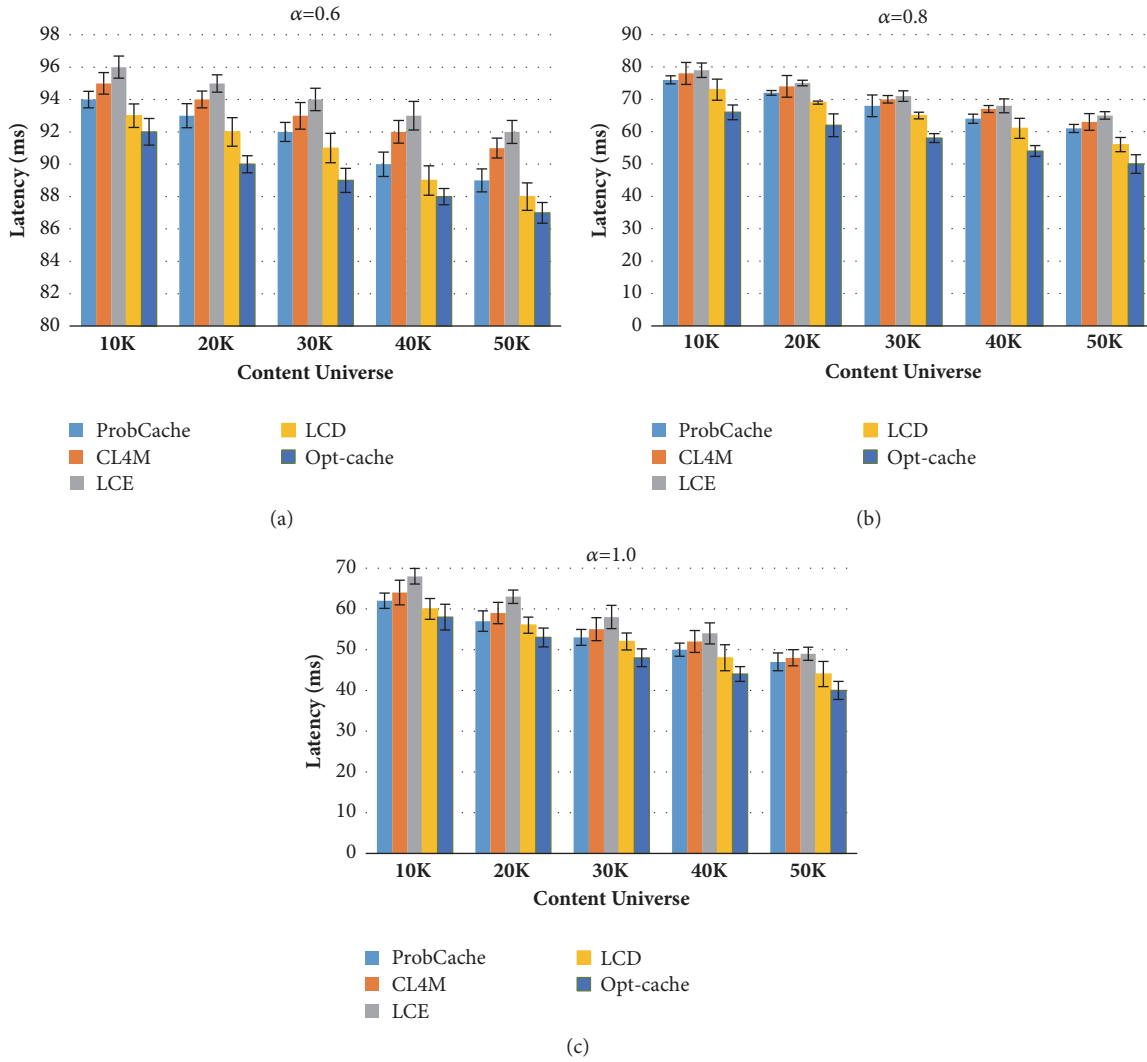
FIGURE 4: Latency performance with varying content universe.

even at lower cache size ratio. Similarly, greater value of skewness parameter indicates more popular content are requested resulting in greater hit ratio. We also observe that although the performance of all strategies at lower popularity model $\alpha=0.6$ is not good, still performance of *opt-Cache* is better than other strategies due to optimal cache placement.

4.3. Discussion on Path Stretch Performance. Although hit ratio provides an indication of the percentage of the requests served within the network, path stretch defines the percentage of the path that a content has travelled to retrieve the content or the ratio between the actual path and the shortest path length. Figures 6(a), 6(b), and 6(c) indicate the path stretch ratio at different popularity models and cache size ratio. We observe that hop count of *opt-Cache* decreases at different popularity models with different cache sizes, and performance is superior to existing techniques. We observe

that *opt-Cache* decreases average hop count from 18 to 51%.

4.4. Discussion on Link Load Performance. Link load is the number of bytes travelled on the link to obtain content from the source. It is an important performance parameter to calculate the bandwidth utilization of the content in the network. We perform simulation for link load at

(a) Low popularity model ($\alpha=0.6$) and high popularity rate ($\alpha=1.0$) at variable cache size (4%–20%). We find a significant improvement in link load performance for proposed scheme. Figures 7(a) and 7(b) show the improvement in the link load performance of proposed scheme.

(b) Variation in popularity model keeping the cache size fixed. Using these settings, we observe the improved performance of *opt-Cache* as shown in Figures 8(a) and
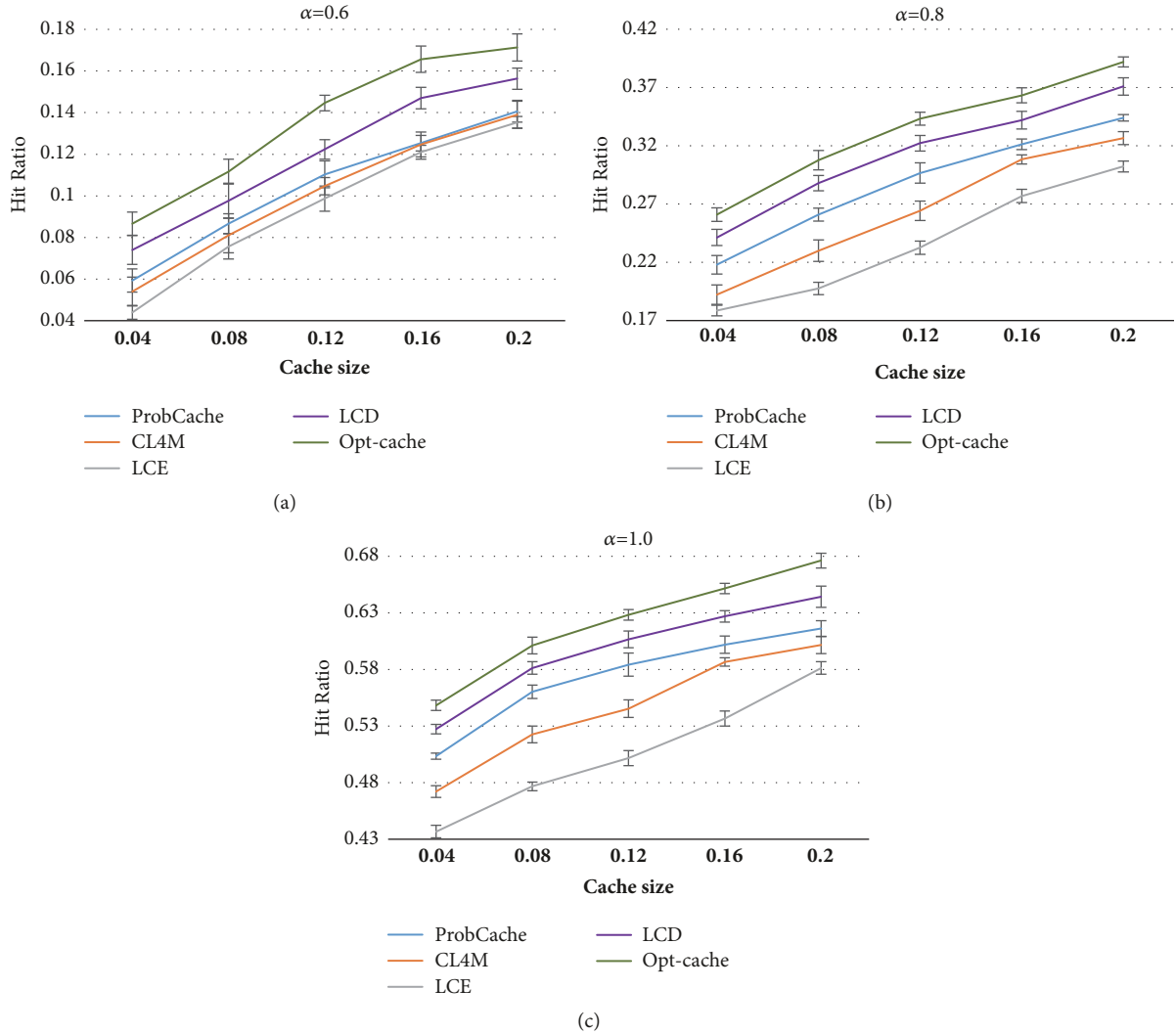
(a)

(b)

(c)

FIGURE 5: Hit ratio with respect to varying cache size and popularity parameter α.

8(b) as the content placement is optimal under such conditions.

5. Conclusions

In this work, we have designed and evaluated a content cache placement scheme for CCN. There are various conflicting challenges in content caching as we increase the cache size the more popular content can be cached; however deployment cost increases. Similarly, if we try to reduce the cache size the deployment cost decreases but lesser amount of popular content can be cached. With limited cache size and a large number of content in the network, an intelligent content cache minimizes bandwidth utilization and latency of content retrieval. Achieving a trade-off between conflicting objectives is a major challenge that has been addressed in our work. Our main contribution is the mathematical model to compute optimal cache placement, independent of any particular simulation framework. Proposed scheme determines optimized

set of content to be cached at each node towards the edge based on content popularity and content distance from the content source. From the simulation study, it is observed that our proposed caching scheme has the potential in enhancing the system performance as compared to existing caching schemes.

In future, we want to explore the content placement based on network flow and centrality parameters, where the nodes with high flow rate and centrality cache the popular content. We also plan to improve our scheme towards dynamic topologies and study the impact of mobility on proposed scheme.

Data Availability

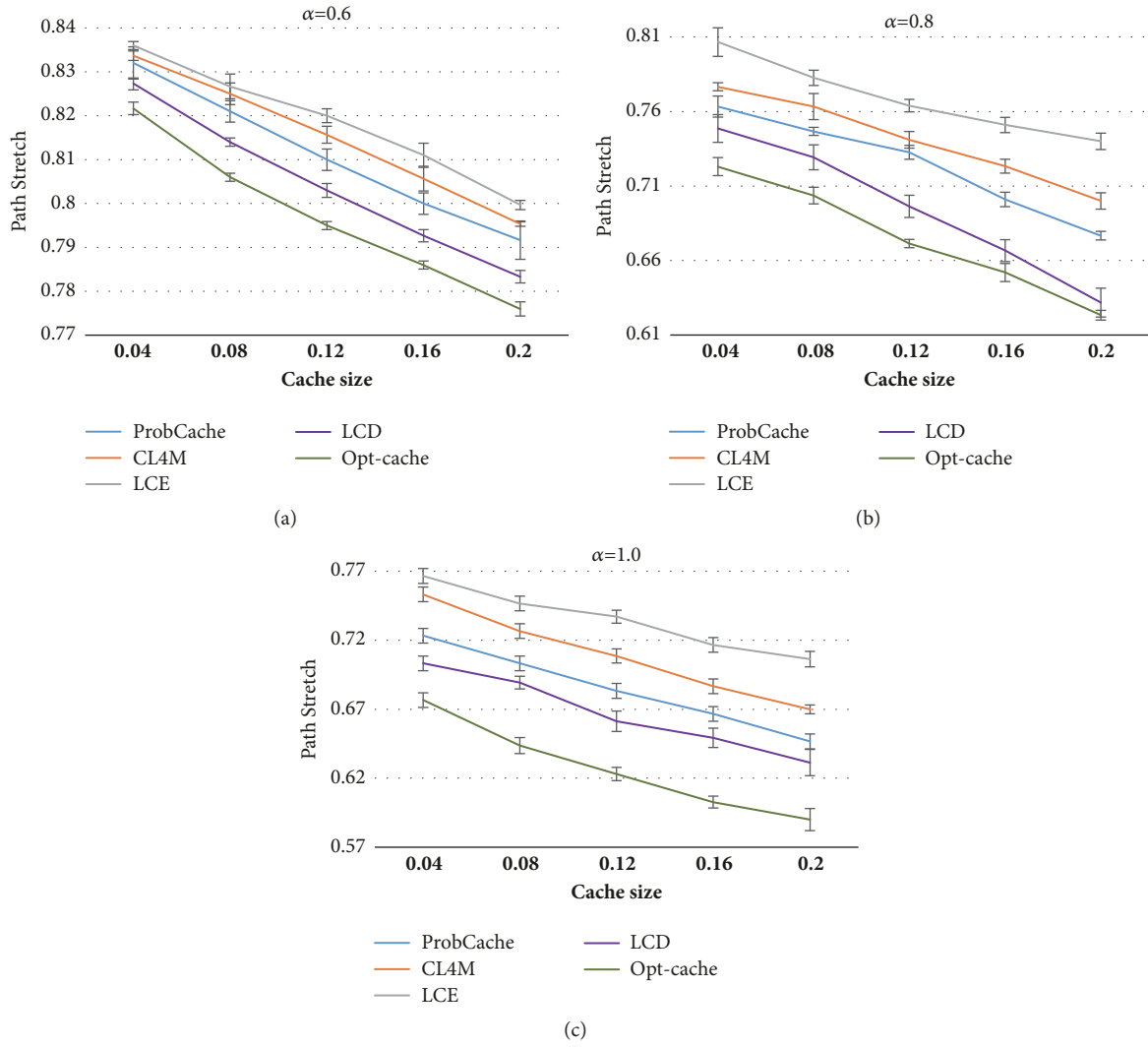All the data related to the results presented in this paper can be found at the following link: https://www.dropbox.com/sh/ekiie1mkwuwnr2r/AACAS6CBjYhwI9SJmhFI0D2na?dl=0.

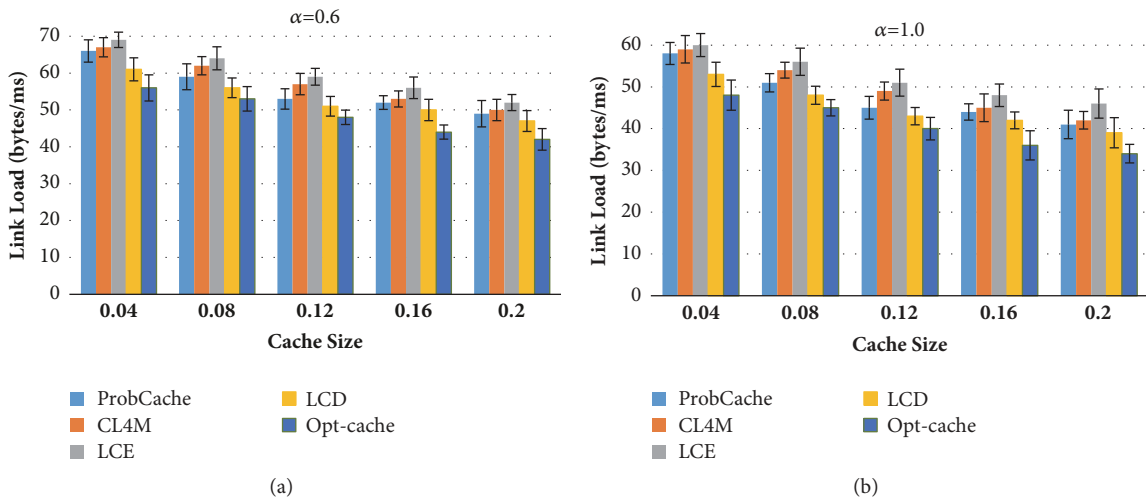FIGURE 6: Path stretch with varying cache size and popularity parameter α.

FIGURE 7: Link load with varying cache size and popularity parameter α.
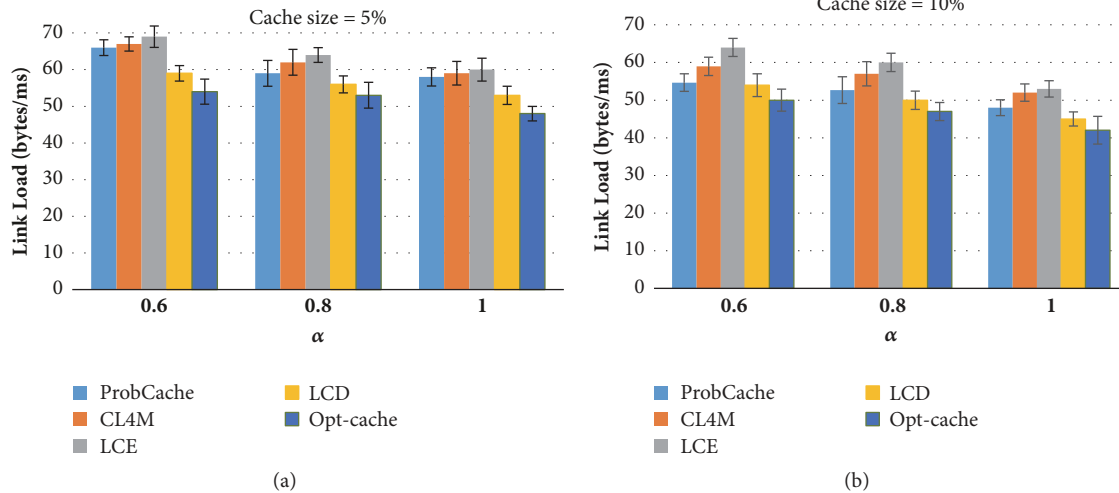
FIGURE 8: Link load with varying values of popularity parameter and fixed cache size.
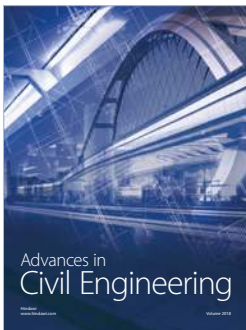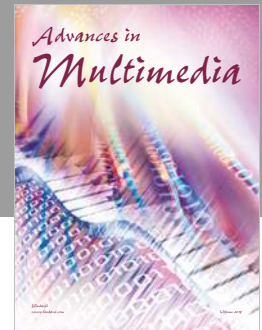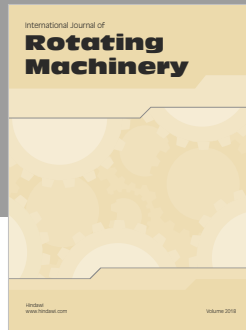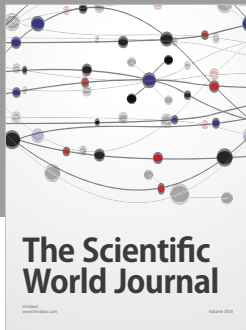
Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. H. Ahmed, S. H. Bouk, and D. Kim, "Content-centric networks: an overview, applications and research challenges," in *SpringerBriefs in Electrical and Computer Engineering*, Springer, Germany, 2016.

[2] Y. Li, T. Lin, H. Tang, and P. Sun, "A chunk caching location and searching scheme in Content Centric Networking," in *Proceedings of the ICC 2012 - 2012 IEEE International Conference on Communications*, pp. 2655–2659, Ottawa, ON, Canada, June 2012.

[3] M. S. Akbar, K. A. Khaliq, R. N. B. Rais, and A. Qayyum, "Taxonomy of Information-Centric Networking: Categorization and Challenges," in *Proceedings of the 23rd Wireless Optical and Communication Conference (WOCC)*, Newark, NJ, USA, 2014.

[4] G. Li, J. Li, and J. Wu, "Fog-enabled Edge Learning for Cognitive Content-Centric Networking in 5G," 2018, https://arxiv.org/abs/1808.09141.

[5] F. Song, Z. Ali, J. Li et al., "Smart Collaborative Caching for Information-Centric, IoT in Fog Computing," *Sensors*, vol. 17, no. 11, 2017.

[6] D. Grewe, M. Wagner, M. Arumaithurai, I. Psaras, and D. Kutscher, "Information-centric mobile edge computing for connected vehicle environments: Challenges and research directions," in *Proceedings of the 1st International Workshop on Mobile Edge Communications, MECOMM 2017*, pp. 7–12, Los Angeles, CA, USA, 2017.

[7] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT '09)*, pp. 1–12, Rome, Italy, December 2009.

[8] D. Kutscher, S. Eum, K. Pentikousis et al., "Information-Centric Networking (ICN) Research Challenges," https://tools.ietf.org/html/draft-irtf-icnrg-challenges-06.

[9] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie, "Design and evaluation of the optimal cache allocation for content-centric networking," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 65, no. 1, pp. 95–107, 2016.

[10] S. Ali, G. Rizzo, V. Mancuso, and M. A. Marsan, "Persistence and availability of floating content in a campus environment," in *Proceedings of the 34th IEEE Annual Conference on Computer Communications and Networks, IEEE INFOCOM 2015*, pp. 2326–2334, Hong Kong, May 2015.

[11] O. Khalid, S. U. Khan, S. A. Madani et al., "Opportunistic databank: A context aware on-the-fly data center for mobile networks," in *Handbook on Data Centers*, pp. 1077–1094, Springer-Verlag, New York, NY, USA, 2015.

[12] S. Naz, R. Naveed Bin Rais, P. A. Shah et al., "A dynamic caching strategy for CCN-based MANETs," *Computer Networks*, vol. 142, pp. 93–107, 2018.

[13] S. Naz, R. N. Rais, and A. Qayyum, "A Resource Efficient Multi-Dimensional Cache Management Strategy in Content Centric Networks," *Journal of Computational and Theoretical Nanoscience*, vol. 15, no. 4, pp. 1137–1152, 2018.

[14] J. M. Wang, J. Zhang, and B. Bensaou, "Intra-AS cooperative caching for content-centric networks," in *Proceedings of the 2013 3rd ACM SIGCOMM Workshop on Information-Centric Networking, ICN 2013*, pp. 61–66, August 2013.

[15] S. Naz, R. N. B. Rais, and A. Qayyum, "Multi-Attribute Caching: Towards efficient cache management in Content-Centric Networks," in *Proceedings of the 13th IEEE Annual Consumer Communications and Networking Conference, CCNC 2016*, pp. 630–633, Las Vegas, Nev, USA, January 2016.

[16] X. Hu, J. Gong, G. Cheng, and C. Fan, "Enhancing in-network caching by coupling cache placement, replacement and location," in *Proceedings of the IEEE International Conference on Communications, ICC 2015*, pp. 5672–5678, June 2015.

[17] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *Proceedings of the 2nd ACM SIGCOMM Information-Centric Networking Workshop, ICN '12*, pp. 55–60, August 2012.

[18] L. Saino, I. Psaras, and G. Pavlou, "Icarus: A Caching simulator for Information Centric Networking (ICN)," in *Proceedings*

of the 7th International Conference on Simulation Tools and Techniques, SIMUTools 2014, pp. 66–75, Portugal, March 2014.

[19] H. K. Rath, B. Panigrahi, and A. Simha, "On cooperative on-path and off-path caching policy for information centric networks (ICN)," in *Proceedings of the 30th IEEE International Conference on Advanced Information Networking and Applications, AINA 2016*, pp. 842–849, March 2016.

[20] Y. He, Y. Zhu, Y. Ni, J. Shi, and N. Zhu, "A Cache Strategy in Content-centric Networks Based on Node's Importance," *Information Technology Journal*, vol. 13, no. 3, pp. 588–592, 2014.

[21] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache "Less for More" in Information-Centric Networks," in *Proceedings of the International Conference on Research in Networking, NETWORKING 2012*, vol. 7289 of *Lecture Notes in Computer Science*, pp. 27–40, Springer, Berlin, Germany, 2012.

[22] G. Carofiglio, L. Mekinda, and L. Muscariello, "LAC: Introducing latency-Aware caching in Information-Centric Networks," in *Proceedings of the 2015 IEEE 40th Conference on Local Computer Networks, LCN 2015*, pp. 422–425, USA, October 2015.

[23] J. Ren, W. Qi, C. Westphal et al., "MAGIC: A distributed max-gain in-network Caching strategy in information-centric networks," in *Proceedings of the IEEE INFOCOM 2014 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 470–475, Toronto, ON, Canada, April 2014.

[24] W. Zhao, Y. Qin, D. Gao, C. H. Foh, and H.-C. Chao, "An Efficient Cache Strategy in Information Centric Networking Vehicle-to-Vehicle Scenario," *IEEE Access*, vol. 5, pp. 12657–12667, 2017.

[25] C. Xu, W. Quan, A. V. Vasilakos, H. Zhang, and G.-M. Muntean, "Information-centric cost-efficient optimization for multimedia content delivery in mobile vehicular networks," *Computer Communications*, vol. 99, pp. 93–106, 2017.

[26] S. Ullah, K. Thar, and C. S. Hong, "Management of scalable video streaming in information centric networking," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21519–21546, 2017.

[27] Y. Kim and I. Yeom, "Performance analysis of in-network caching for content-centric networking," *Computer Networks*, vol. 57, no. 13, pp. 2465–2482, 2013.

[28] C. Bernardini, T. Silverston, and O. Festor, "A comparison of caching strategies for content centric networking," in *Proceedings of the 58th IEEE Global Communications Conference, GLOBECOM 2015*, USA, December 2015.

[29] N. Laoutaris, H. Che, and I. Stavrakakis, "The LCD interconnection of LRU caches and its analysis," *Performance Evaluation*, vol. 63, no. 7, pp. 609–634, 2006.

[30] C. Xu, W. Quan, H. Zhang, and L. A. Grieco, "GrIMS: Green Information-Centric Multimedia Streaming Framework in Vehicular Ad Hoc Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 2, pp. 483–498, 2018.

[31] W. Quan, C. Xu, J. Guan, H. Zhang, and L. A. Grieco, "Social cooperation for information-centric multimedia streaming in highway VANETs," in *Proceedings of the 15th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2014*, Sydney, NSW, Australia, 2014.

[32] W. Quan, C. Xu, A. V. Vasilakos, J. Guan, H. Zhang, and L. A. Grieco, "TB2F: Tree-bitmap and bloom-filter for a scalable and efficient name lookup in Content-Centric Networking," in *Proceedings of the 2014 IFIP Networking Conference, IFIP Networking 2014*, Trondheim, Norway, June 2014.

[33] W. Quan, C. Xu, J. Guan, H. Zhang, and L. A. Grieco, "Scalable name lookup with adaptive prefix bloom filter for named data networking," *IEEE Communications Letters*, vol. 18, no. 1, pp. 102–105, 2014.

[34] S. M. Azimi, O. Simeone, A. Sengupta, and R. Tandon, "Online edge caching and wireless delivery in fog-aided networks with dynamic content popularity," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1189–1202, 2018.

[35] M. A. Maddah-Ali and U. Niesen, "Fundamental Limits of Caching," in *Proceedings of the International Symposium on Information Theory*, Bell Labs, Alcatel-Lucent, 2013.

[36] M. Femminella, G. Reali, D. Valocchi, R. Francescangeli, and H. Schulzrinne, "Advanced caching for distributing sensor data through programmable nodes," in *Proceedings of the 2013 19th IEEE Workshop on Local and Metropolitan Area Networks, LANMAN 2013*, Brussels, Belgium, April 2013.

[37] G. Reali, M. Femminella, E. Nunzi, and D. Valocchi, "Genomics as a service: A joint computing and networking perspective," *Computer Networks*, vol. 145, pp. 27–51, 2018.

[38] J. Li, B. Liu, and H. Wu, "Energy-efficient in-network caching for content-centric networking," *IEEE Communications Letters*, vol. 17, no. 4, pp. 797–800, 2013.

[39] S. Wang, X. Huang, Y. Liu, and R. Yu, "CachinMobile: An energy-efficient users caching scheme for fog computing," in *Proceedings of the 2016 IEEE/CIC International Conference on Communications in China, ICCC 2016*, China, July 2016.

[40] R. Wang, X. Peng, J. Zhang, and K. B. Letaief, "Mobility-aware caching for content-centric wireless networks: Modeling and methodology," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 77–83, 2016.

[41] J. A. Khan, C. Westphal, and Y. Ghamri-Doudane, "A content-based centrality metric for collaborative caching in information-centric fogs," in *Proceedings of the 2017 IFIP Networking Conference (IFIP Networking) and Workshops*, pp. 1–6, Stockholm, Sweden, June 2017.