Computers, Materials & Continua
DOI:10.32604/cmc.2022.020587
Article

Tech Science Press

COCP: Coupling Parameters Content Placement Strategy for In-Network Caching-Based Content-Centric Networking

Salman Rashid[1], Shukor Abd Razak[1], Fuad A. Ghaleb[1,*], Faisal Saeed[2] and Eman H. Alkhammash[3]

[1]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru, 81310, Malaysia
[2]College of Computer Science and Engineering, Taibah University, Medina 42353, Saudi Arabia
[3]Department of Computer Science, College of Computers and Information Technology, Taif University, Taif, 21944, Saudi Arabia
*Corresponding Author: Fuad A. Ghaleb. Email: abdulgaleel@utm.my
Received: 15 June 2021; Accepted: 30 July 2021

Abstract: On-path caching is the prominent module in Content-Centric Networking (CCN), equipped with the capability to handle the demands of future networks such as the Internet of Things (IoT) and vehicular networks. The main focus of the CCN caching module is data dissemination within the network. Most of the existing strategies of in-network caching in CCN store the content at the maximum number of routers along the downloading path. Consequently, content redundancy in the network increases significantly, whereas the cache hit ratio and network performance decrease due to the unnecessary utilization of limited cache storage. Moreover, content redundancy adversely affects the cache resources, hit ratio, latency, bandwidth utilization, and server load. We proposed an in-network caching placement strategy named Coupling Parameters to Optimize Content Placement (COCP) to address the content redundancy and associated problems. The novelty of the technique lies in its capability to minimize content redundancy by creating a balanced cache space along the routing path by considering request rate, distance, and available cache space. The proposed approach minimizes the content redundancy and content dissemination within the network by using appropriate locations while increasing the cache hit ratio and network performance. The COCP is implemented in the simulator (Icarus) to evaluate its performance in terms of the cache hit ratio, path stretch, latency, and link load. Extensive experiments have been conducted to evaluate the proposed COCP strategy. The proposed COCP technique has been compared with the existing state-of-the-art techniques, namely, Leave Copy Everywhere (LCE), Leave Copy Down (LCD), ProbCache, Cache Less for More (CL4M), and opt-Cache. The results obtained with different cache sizes and popularities show that our proposed caching strategy can achieve up to 91.46% more cache hits, 19.71% reduced latency, 35.43% improved path stretch and 38.14% decreased link load. These results confirm that the proposed COCP strategy has the potential capability to handle the demands of future networks such as the Internet of Things (IoT) and vehicular networks.

Keywords: Content-centric networking; on-path caching; content redundancy; security; privacy; data dissemination; internet of things

1 Introduction

The Internet has an important role in our daily life. It has been designed to meet the requirements of the communication system. The frequent transmissions of the same content on the Internet generate many issues due to the incredible growth of Internet traffic [1]. For example, these issues include increased server load, greater bandwidth requirements, higher latency rate, and unnecessary usage of cache resources. Content-Centric Networking (CCN) is an active area of research in future internet architecture and next-generation networks and is one of the key protocols being explored for the Internet of Things (IoT). Internet architectures based CCN is better than host-centric architecture as it is capable to handle future network applications and big data demands [2]. Although, TCP/IP does not fit properly in IoT. CCN, however, is considered an alternative to TCP/IP for IoT. CCN assigns a unique name to data and addresses data objects at the network level, which is a major difference from the Internet's host-centric architecture. CCN naturally supports many-to-many communication and in-network caching [3]. The network tries to answer the request with a data object when a request from a user application carries the name or identification of the required data object. The name can also refer to a host machine or a location, making CCN more general than the host-to-host communication model [4].

CCN permits the node to store a temporary copy of the requested content when a request is made from the source on the requested path [5]. CCN reduces the response time by providing content availability closer to the user. CCN also minimizes the server load, network traffic, and maximizes the availability of desired contents [6]. However, it is a complex problem to decide, which content will be cached at which location to produce better and efficient results in terms of caching and network performance [7].

In-network caching has many challenges including limitations of storage, caching i.e., replacement and placement, caching traffic, and network topology complexity [8,9]. Misuse of limited cache capacity adversely affects in-network caching performance, which affects content diversity, latency, path stretch, bandwidth, and the cache hit ratio [10–12]. Content placement policy decides which content should be cached in the appropriate place to improve the cache hit ratio and network performance [13]. Numerous caching strategies have been developed for content placement to manage the CCN in-network caching, such as Leave Copy Everywhere (LCE) [14], Leave Copy Down (LCD) [15], ProbCache [16], Cache Less for More (CL4M) [17], and opt-Cache [8]. There is no optimal solution of content cached along with downloading path, therefore, it causes redundancy, underutilized cache space, and low diversity of contents. Consequently, cache resources will be wasted by holding identical items, which degrades the performance of overall networks.

In this research, a novel technique Coupling Parameters to Optimize Content Placement (COCP) for content caching is proposed to improve content dissemination and minimizes content redundancy within the network. This mechanism utilizes the neighbor information for low content retrieval delay and selects a suitable location for content placement by using different nature parameters for a better cache hit ratio, latency, path stretch, and link load. The novelty of the proposed technique is to utilize joint information of request rate, distance, and available cache space to optimize the node selection. The main contribution is summarized as follows:

- The proposed mechanism named Coupling Parameters to Optimize Content Placement (COCP) minimizes the content redundancy and content dissemination through optimized node selection to improve the cache hit ratio, latency, path stretch, and link load.

- A mechanism for better network performance is designed and developed in this study by coupling different parameters such as joint request rate, distance, and available cache space to minimize content redundancy.
- The Icarus simulator [18] was used in this study for evaluating the performance of the proposed model as compared with the existing state-of-the-art algorithms and also verify the impact of different parameters such as content popularity, path stretch, and cache size. Subsequently, we gain substantial improvement in cache hit ratio, latency, path stretch, and link load.

This paper is structured as follows. Section 2 contains previous work, placement policy is defined in Section 3, simulation setup is defined in Section 4 whereas the result analysis and discussion with performance metrics are available in Section 5. The conclusion is described in Section 6.

2 Related Work

In the caching strategy based on default CCN, when a request to download a user content has been received, a copy of the requester's content is stored on all the nodes along the publisher-consumer path. In-network caching plays an important role in content distribution. A router in the path can cache content locally, which allows it to deal with subsequent requests of the same content [19]. Hence, on-path caching reduces communication and computation overhead during the transmission and insertion of data in the network [7,20]. If the immediate router is lacking an interest packet, it will plan the customer's request in the network and arrange the required content for the customer. Content caching criteria depends on various parameters. For example, caching decisions can depend on content popularity, latency, probabilistic positions, the centrality of the network, or distance from the requester to a source.

LCE [14] is a default cache placement policy that caches a copy of the content on each router along the delivery path. Content must be cached on the router along the delivery path, as the same content cached at the neighbor causes redundancy, underutilized cache space, and low diversity of contents. This technique is highly redundant due to its nature and inefficient cache utilization within CCN interconnections; hence it requires a large cache storage capacity. Furthermore, there are no criteria for defining the appropriate node selection for content caching. However, it has higher bandwidth utilization, latency, path stretch (hop count), and a lower hit ratio.

CL4M [17] diminishes content repetition by utilizing the idea of centrality to decide the router generally fit to cache content on the way between the source and client. This technique utilizes the calculation of the centrality of the router based on the number of delivery paths. It caches the content in the router which has the highest centrality in the delivery path. This concept is very useful for in-network content caching because most intersections pass at these routers that have the highest centrality. That means CL4M has capable of reducing the content redundancy. The cost of caching operations is very high at the selected node due to centrality.

The aim of ProbCache [21] is to provide fairness about cache storage along the delivery path over different contents. This policy results in inefficient usage of cache resources because redundant information is cached in the delivery path at more than one node. Due to redundant data, replacement operations are high in the small publisher to consumer path. ProbCache could not achieve the goal of fair allocation of the cache capacity along the delivery path. It is not capable to provide the positioning of the node distinction on behalf of the cache capacity. ProbCache makes no distinction on what content to cache. Therefore, ProbCache fails to address the problem of content redundancy, which results in decreasing the cache hit ratio and increasing the latency.

Another technique, opt-Cache, is the most well-known content placement technique in the CCN [8]. It is based on popularity and content distance from the generator. The objective of this technique is to reduce the utilization of bandwidth and latency by using content placement with diminished content redundancy. If the content is popular, then more time is allocated to stay in the cache. Every content is attached with CGP (Cache Gain Popularity) value, which is calculated by the request rate (popularity) and hop count (distance) for every content. When a hit occurs, the value of CGP(f) is updated. If one content has more hits than other content, its CGP(f) will be higher than the other. In opt-Cache, when a request is generated from leaf, then the server will reply to the consumer, and each router along the path will hold content in its cache. After some time, the cache will be full due to content redundancy. Thus, the router is not able to serve its user request efficiently, which results in underutilized cache resources.

These caching strategies increase the homogeneous content replications through caching similar contents at multiple nodes that create a high level of content redundancy and reduces the content diversity which affects the cache hit ratio, latency, path stretch, and link load. They do not use request rate, distance, and available cache space jointly for suitable node selection with balancing between available cache space along the routing path. The equilibrium between these parameters is highly important for achieving the overall better performance of network and cache resources keeping in view the limited cache space. It is pertinent to mention here that the attention of researchers has been limited to optimizing the selected node for caching using different parameters jointly. From the literature, we can conclude that the existing problems can be minimized through an efficient caching policy. We proposed an in-network caching policy for CCN that minimizes redundancy, latency, bandwidth utilization, and increases the cache hit ratio by taking into account cache size, the distance between publisher to consumer, and content request rate. We proposed COCP to solve the above-mentioned problems by using distance, request rate, and space jointly.

3 Proposed Content Placement Strategy

In this section, we present a detailed system model and design of COCP, which minimizes the content redundancy and optimizes the performance of the cache in CCN. Therefore, the same content object does not cache more than once along the routing path by using the relationship of different parameters for selecting the best suitable location. The necessity of different research approaches to verify and validate CCN research results from using different widely used platforms.

3.1 System Model

This section presents the overall architecture of the proposed scheme and its major components. We considered the network as a connected graph $G = (V,\ E)$ where V represents the set of nodes $V = \{v_1, v_2, v_3, \ldots, v_n\}$ and E is the set of links $E = \{e_1, e_2, e_3, \ldots, e_m\}$. Let $P = \{p_1, p_2, p_3, \ldots, p_n\}$ be the set of publisher in the network, $\psi = \{\psi_1, \psi_2, \psi_3, \ldots, \psi_n\}$ be the set of maximum storage capacity in the node, $F = \{f_1, f_2, f_3, \ldots, f_n\}$ is the set of contents available in the network. The content request is generated by the CCN user (consumer) and forwarded to the adjacent connected router. For instance, when a user generates an interest packet for the specific content, the CCN router receives the interest packet and searches in the Content Store (CS). If a cache hits, then a reply is sent to the requested node; otherwise, the request is forwarded to the next connected router. CCN routers are standard routers that have more caching capacity. However, this extra storage helps to cache content in the local cache.

Fig. 1 demonstrates the three data structures (Forwarding Interest Base (FIB), Pending Interest Table (PIT), and Content Store (CS)) that are used in each CCN router [22]. FIB is similar to the IP

forwarding table. It is responsible to forward interest requests to probable sources [23]. The FIB drops the requested packet when it has no information about the next hop. PIT keeps the information in tabular format about unexecuted interests. Each record in the PIT contains a list of incoming interfaces with the content name. PIT works like a data structure similar to the hash table. Every entry in the PIT shows the source that keeps track of the interest packet forwarded upstream towards the content provider. As soon as the data packet is received from the publisher, it is sent towards the downstream node using the same entry [24]. Data packets are temporarily placed in CS [25]. Replacement of data packet is governed by the implementation of eviction policy i.e., Least Frequently Used (LFU) or Least Recently Used (LRU). Reused data increases due to eviction policies [26]. CCN domain consists of two types of sources (server, router). Contents that are published from CCN servers are temporarily stored in a local cache (CS).

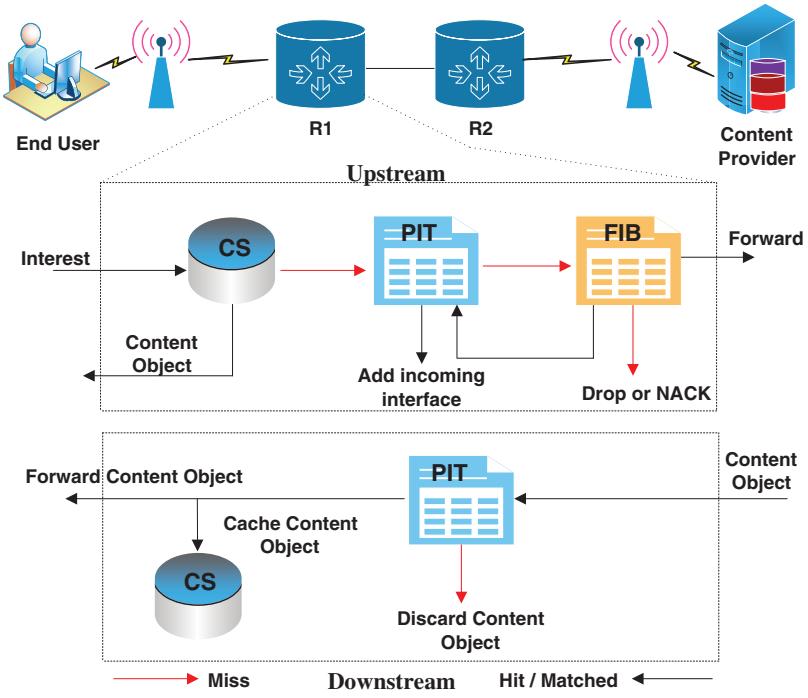

Figure 1: Forwarding process at CCN node

3.2 COCP Strategy

Numerous content caching approaches were introduced to make better use of in-network caching resources. Content redundancy has produced many problems such as minimizing content diversity, low hit ratio, high latency, and bandwidth wastage. It is difficult to solve these problems under limited cache size and random request constraints. The cache performance is dependent on caching the content at a suitable location with low redundancy. Several content placement policies have been developed through different parameters, such as popularity, latency, centrality, and hop-count. The COCP improved the CCN-based content caching method by using different parameters including request rate, the distance between consumer to the publisher, and available cache space simultaneously. Furthermore, the content is cached at a suitable place to improve the cache hit ratio and network performance.

Algorithm 1: The Proposed COCP Strategy

Input:
Request for content f_i
Output:
Content placement at the most appropriate place

```
For each hop in the path
  if the cache hit at Node v then
    serving_node ← v
  else
    Neighbors ← All v neighbors
    for NNode in Neighbors
      if the cache hit at Node NNode then
      serving_node ← NNode
      break
    end for
    serving_node ← P
  end if
end for
```

$$LProb(f_i) \leftarrow \frac{\psi(v_i)}{\sum_{k=1}^{n}\psi(v_k)} \times \frac{\lambda(f_i, v_j)}{\sum_{k=1}^{n}\lambda(f_k, v_j)} \times \frac{D(f_i, v_j)}{\bar{D}(C, P)}$$

```
for each hop in the path
      if ψ(v_i) > 0 and min{LProb_i} then
          cache content f_i
        break
    end if
  end loop
```

Numerous In algorithm 1, the node v received an interest packet from the consumer for f_i. Node v retrieves f_i from CS then assembles the data packet for the reply and discards the interest packet. If the f_i is not present in CS, it tries to find the content from node v neighbors. If the content is not found at neighbors, the interest packet is then forwarded to the next node using the information obtained from FIB. Node v keeps unresolved forward interest packet information in a PIT that collects packets of similar interest from the different consumers. If f_i is found at the publisher (P) or along the routing path, the *LProb* value is calculated. Therefore, the lower *LProb* value helped to select the most suitable location for caching.

$$LProb(f_i) \leftarrow \frac{\psi(v_i)}{\sum_{k=1}^{n}\psi(v_k)} \times \frac{\lambda(f_i, v_j)}{\sum_{k=1}^{n}\lambda(f_k, v_j)} \times \frac{D(f_i, v_j)}{\bar{D}(C, P)} \tag{1}$$

$\psi(v_i)$ represents the available cache space at a particular node v and $\sum_{k=1}^{n}\psi(v_k)$ shows the accumulative cache size along the routing path. It is $\frac{\psi(v_i)}{\sum_{k=1}^{n}\psi(v_k)}$ representing the normalized value of the available node capacity. If the normalized value approaches zero, it represents the low cache capacity. $\lambda(f_i, v_j)$ represents the request rate for a specific content f_i at node v_j. The $\sum_{k=1}^{n}\lambda(f_k, v_j)$ represents the total request rate at a particular node. Distance traversal of the f_i found from consumer to the serving node is calculated by $D(f_i, v_j)$. Traversal of the f_i to reach from consumer to publisher is done by $\bar{D}(C, P)$

which is the maximum path stretch. When content is available far from the requester or publisher, $LProb(f_i)$ value is then calculated. If the *LProb* value is minimum and the capacity of the node is available, then it is the best node for caching to the particular content f_i.

Fig. 2 illustrates the basic concept of the COCP content caching mechanism. There are publishers A and B that publish the contents GC, RC, and BC on R1 in this network diagram. Furthermore, consumers A, B, C, D & E are connected to R6, R10, R11, R9, R2, respectively, to generate the interest packets for their desired contents. As shown, consumer A is generating a BC interest packet which is received by R6. Subsequently, R6 starts the process of look-up in CS for BC content. If R6 finds the desired content from CS, it will directly respond to consumer A through the data packet. In case the required content does not find in CS, it will find the shortest path towards the provider, which is R1, and forward the interest packet to its next router, which is R4 in this scenario. If R4 does not find the desired content, then it will repeat the same act as performed by R6 and send the request to its next router. This process will continue until the required content is found. As the BC was found on R1, it will be returned to its source, which is consumer A, by utilizing the same downloading path. Each router that existed on this path will calculate *LProb* value based on cache size, distance, content request rate decided to cache the BC against the lowest value of *LProb*. Assuming that sufficient contents are cached within the network and consumer C requests for GC. First, R9 will check the information of desired content (GC) with its directly connected neighbors; if GC is not found from them, R9 will forward the request to its next router (R8) by applying the shortest path to the provider. R8 will check for GC from its directly connected neighbors except those already visited by R9. As GC was found from R2, the data packet will be forwarded to consumer C through the same downloading path after calculating the *LProb* value by each router and cache the lowest. COCP significantly improved cache hit ratio, latency, bandwidth utilization, and diversity of contents, whereas path stretch and content redundancy decreased decisively.

4 Simulation Setup

We have described the simulation parameters in this section. We have used the simulation tool Icarus [18], as this simulator has been specifically designed to analyze caching and routing policies in CCN. The four main building blocks of Icarus are experimental orchestration, execution of the experiment, scenario generation, and collection of results. Thus, each time an event is reported, the appropriate timestamps, receiver, and source of the event are saved. The results gathering unit of the simulator collects the simulation result. The delay of each link passed during the content download is calculated with Icarus.

We have set the number of warm-ups and measured requests at 40,000. Warm-up (number of content requests generated to pre-populate the caches) requests are used to settle caches before running the real experiment. Measured requests are the number of content requests after warm-up requests. The popularity of the content is distributed by Zipf's law with the popularity distribution of exponent alpha (α) $\in$ [0.6, 0.8, 1.0]. Moreover, network cache capacity varies from 4%–20% of the total content population. GEANT topology, which comprises 40 nodes and 60 edges. A wide range of experiments is carried out using this topology. Tab. 1 describes the parameters of our simulation setup.

5 Results Analysis and Discussion

To show the effectiveness of our proposed technique, we have compared our simulation results with state-of-the-art existing caching policies including ProbCache, LCE, opt-Cache, LCD, and CL4M. The above schemes are chosen for their importance and consistent effectiveness, as shown in the

literature. We applied various performance metrics such as cache hit ratio, latency, link load, and path stretch to show effectiveness in all areas. We have compared these performance matrices one by one as follows:

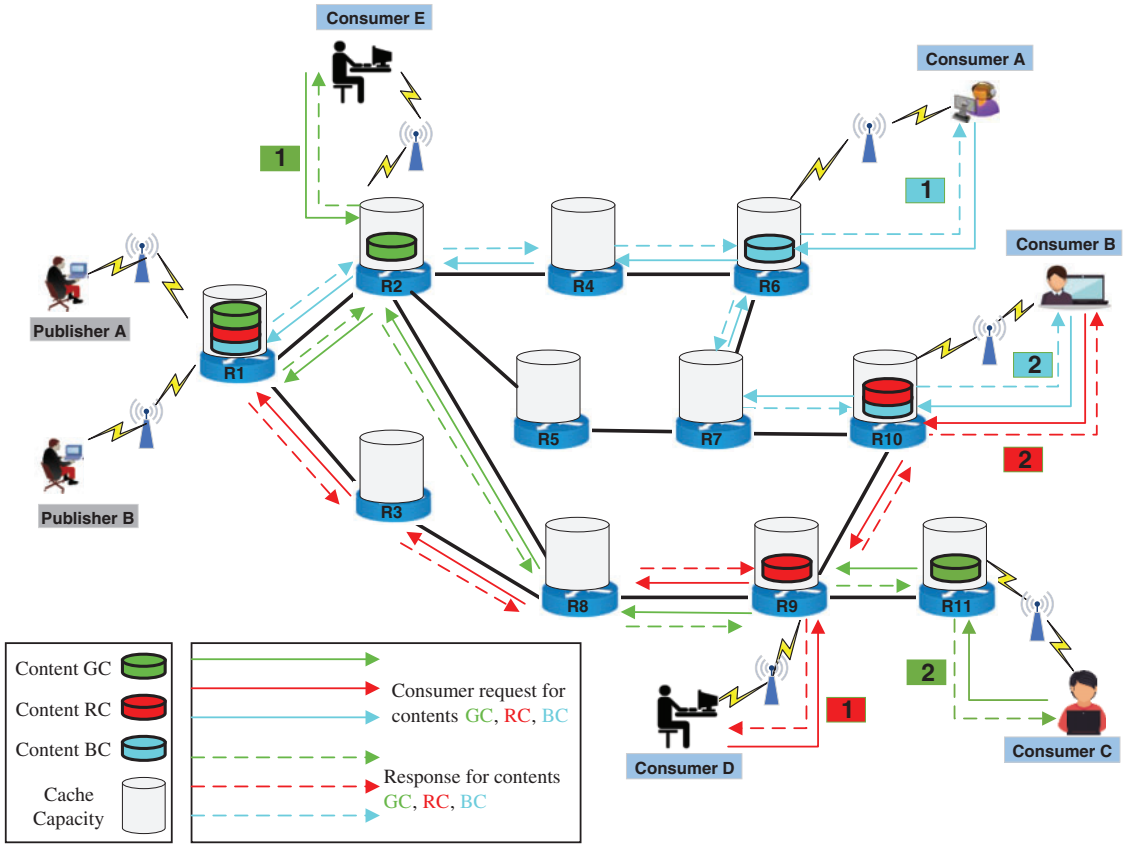

Figure 2: Coupling parameters to optimize content placement

Table 1: Simulation parameters

Parameters	Value
Warm-up requests	40,000
Measured requests	40,000
Model of popularity	0.6, 0.8, 1.0
Total contents	100,000
Cache capacity	4%–20%
Consumer request rate	1.0 request/s
Replacement policy	Least replacement policy (LRU)
Topology	GEANT

5.1 Cache Hit Ratio

The cache hit ratio is a key metric to evaluate the CCN cache performance. It points out the response of in-network cache storage, in which content is stored locally within a specific time limit. When a consumer requested content is available in the cache, it is called the cache hit; otherwise, it is called the cache miss. Content is retrieved from the server when it is not available in the cache. A higher hit ratio reflects the efficiency of the cache and also represents the low server load and low latency, defined as follows:

$$Hit\ Ratio = \frac{Cache_{hits}}{Cache_{hits} + Server_{hits}} \tag{2}$$

The proposed strategy has been compared with existing popular strategies in terms of the cache hit ratio. We simulated different cache capacities and low to high popularity rates. Figs. 3a–3c illustrate that the performance of COCP is better than existing strategies. LCE, CL4M, ProbCache, LCD, and opt-Cache cache hit ratio is low due to replication of similar information along the routing path. It is difficult to cache the new contents within the network when the path is small. Cache space filled up with a low diversity of contents. Hence, some contents evict from the cache to accommodate new content. However, the cache hit ratio decreases due to consumer's new requests accomplished by the publisher. The COCP minimizes the content redundancy which provides more content to be cached along the routing path. As expected, COCP outperformed other caching schemes because it does not waste cache space for storing redundant contents in CCN. COCP performed better specifically in the case of limited cache size. COCP outperformed LCE, CL4M, ProbCache, LCD, and opt-Cache in terms of the cache hit ratio by 91.46%, 39.13%, 20.63%, 8.79%, and 5.79% respectively.

The combination of different parameters used jointly affects the cache hit ratio as shown in Tab. 2. We observed that the cache hit ratio is better when the cache space parameter is used with distance and request rate. However, when the distance and request rate parameters are used jointly, the cache hit ratio is always low because these parameters consider only the distance between the publisher to the consumer with the content request rate. Cache space not fully utilized because there is no parameter to evaluate the effect of the available cache space. Hence, these parameters have no impact on cache space. In short, we jointly used three parameters (Distance, Request rate, and Space) that produced better performance than the combination of the others because these parameters optimize the cache space balancing along the routing path by considering the distance between publisher to consumer and content request rate.

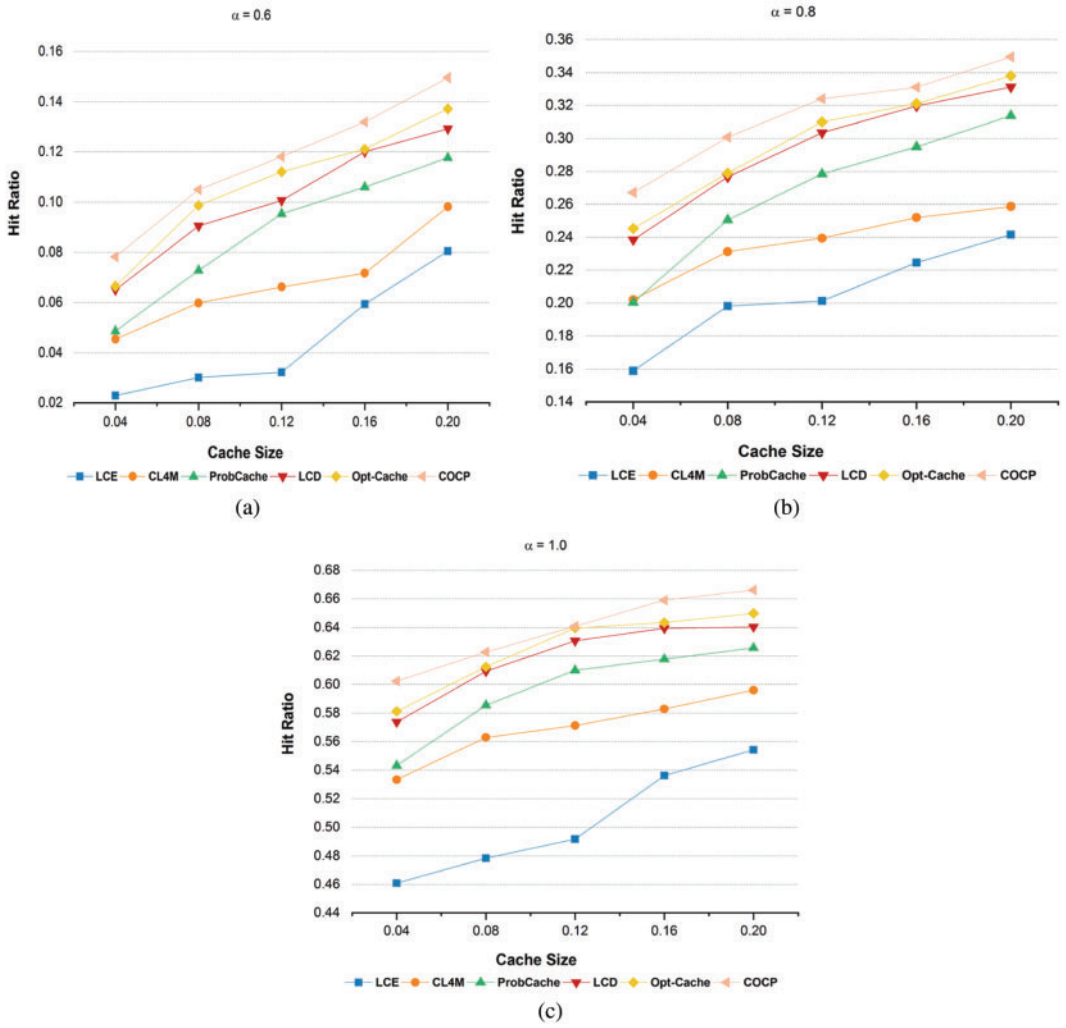

Figure 3: Cache hit ratio with different cache sizes and α (a) Cache hit ratio with different cache sizes and $\alpha = 0.6$ (b) Cache hit ratio with different cache sizes and $\alpha = 0.8$ (c) Cache hit ratio with different cache sizes and $\alpha = 1.0$

Table 2: Effects of different parameters in the cache hit ratio

Parameters	Distance space	Request distance	Request space	Distance request space
	Cache size			
ALFA (α)	**0.04**	**0.04**	**0.04**	**0.04**
0.6	0.072	0.057	0.070	0.078
0.8	0.260	0.250	0.264	0.267
1	0.594	0.584	0.595	0.602
	0.08	**0.08**	**0.08**	**0.08**

(Continued)

Table 2: Continued

Parameters	Distance space	Request distance	Request space	Distance request space
0.6	0.100	0.085	0.102	0.105
0.8	0.292	0.272	0.292	0.301
1	0.617	0.605	0.618	0.623
	0.12	**0.12**	**0.12**	**0.12**
0.6	0.109	0.085	0.104	0.118
0.8	0.318	0.298	0.321	0.324
1	0.635	0.631	0.636	0.641
	0.16	**0.16**	**0.16**	**0.16**
0.6	0.122	0.104	0.120	0.132
0.8	0.325	0.307	0.323	0.331
1	0.640	0.638	0.636	0.659
	0.2	**0.2**	**0.2**	**0.2**
0.6	0.135	0.121	0.135	0.150
0.8	0.339	0.328	0.333	0.350
1	0.652	0.648	0.654	0.666

5.2 Path Stretch (Hop Count)

Path stretch denotes the distance traveled by a consumer interest toward the content provider. When a user generates interest for specific content, the interest packet covered the distance from the consumer to the publisher or a hit occurs from the network node. If the copy of required content is found from the path, then the path stretch value is low; otherwise, the value of the path stretch will be high. The lower path stretch value indicates that the interesting content has been recovered from the consumer neighborhood. Path stretch value reflects the ratio between the shortest path and the actual path length. So, it is necessary to reduce the maximum number of hops between consumer and provider by applying a better caching mechanism.

Figs. 4a–4c show that the performance of the COCP is superior to other existing strategies in terms of path stretch. COCP caches the content closer to the requester at a suitable location. From this location, a requester can retrieve their desired content. Moreover, the content cached location satisfies the reduction in the distance between provider and requester. COCP minimizes the content redundancy and selects the suitable location for caching by jointly considering the request rate, the distance between publisher to the consumer, and available cache space along the routing path by using neighbor information. Therefore, contents are available at appropriate locations along the routing path.

As shown in Fig. 4, COCP achieves a better path stretch with different cache sizes. We observed that LCE, CL4M, and ProbCache performed poorly due to high content redundancy along the routing path and low diversity of contents within the network. Hence, more requests are accommodated by the publisher. In LCD, the user gets the requested content and the publisher keeps a copy of the sent

content only at the neighbor node. It will take a long time to reach content at the edge node or near to the consumer. However, it leads to a higher path stretch value. opt-Cache is performed better due to the high requested content cached for a long time with the distance between publisher to the consumer. When a request is generated from the leaf, the publisher will reply to the consumer, and each router along the path will hold content in its cache. When the cache is full, then it is very hard to cache new contents in which the CGP value of every new content is lower than the older one. Hence, most of the consumers will be satisfied with the publisher instead of the network. COCP outperformed LCE, CL4M, ProbCache, LCD and opt-Cache in terms of path stretch by 35.43%, 31.61%, 30.12%, 28.41% and 21.66%, respectively.

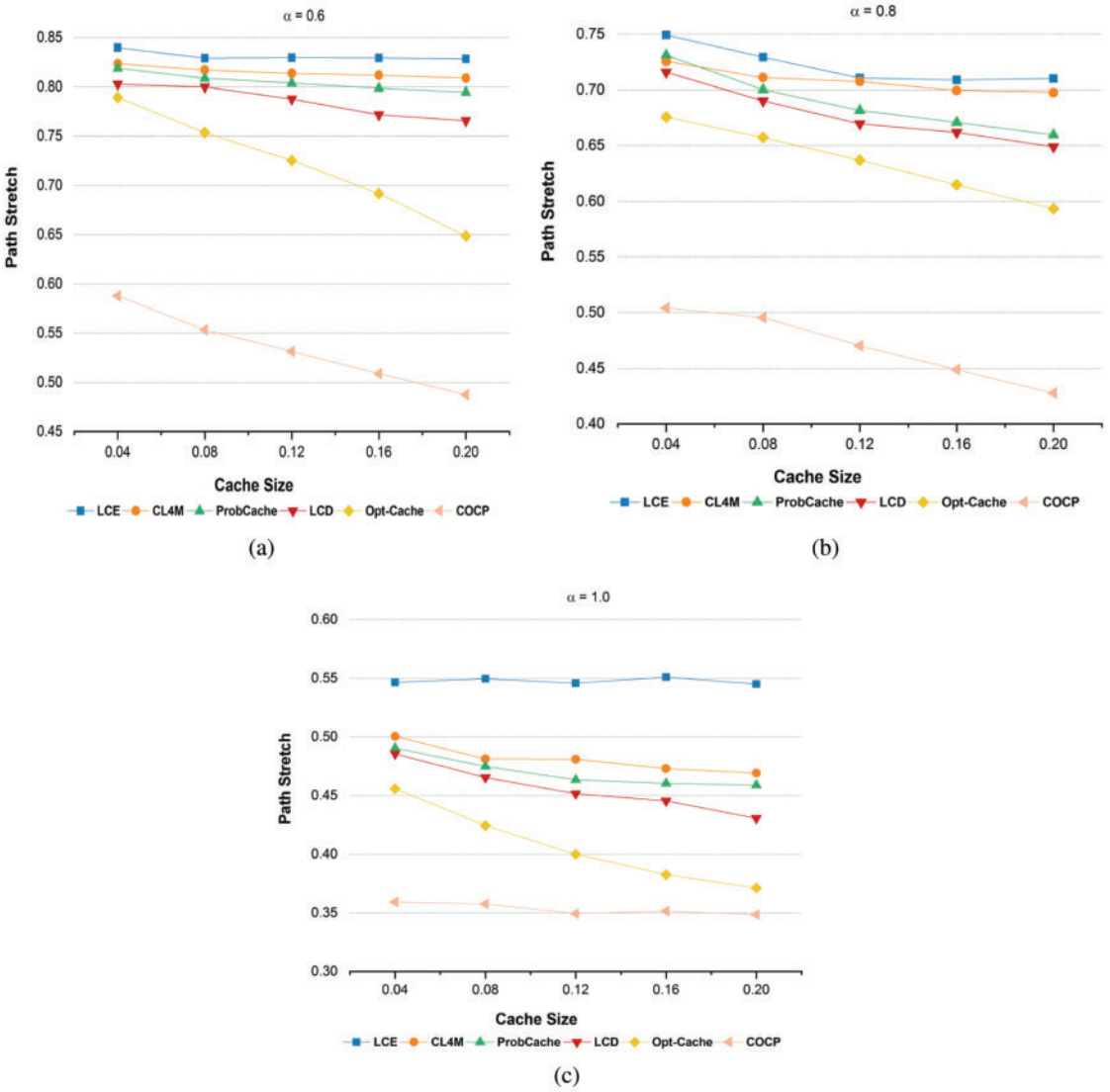

Figure 4: Path stretch with different cache sizes and α (a) Path stretch with different cache sizes and $\alpha = 0.6$ (b) Path stretch with different cache sizes and $\alpha = 0.8$ (c) Path stretch with different cache sizes and $\alpha = 1.0$

Tab. 3 shows the combination of different parameters and their effect at path stretch. When we used the request rate with distance, then the path stretch is high because the content is cached near to the user, but the available cache space was not taken into account. Thus, the eviction process will be high because the edge node filled up the cache space quickly. However, this combination of parameters does not fully utilize the cache space along the routing path. The combination of distance/request rate with space produced better results than the previous combination. These combinations take care of space by consumer request rate and distance between consumer to the publisher. The last combination (distance, request rate, and available cache space) surpassed the previous combinations. This combination assists in deciding the appropriate node selection according to the distance and request rate by considering the available cache space along the routing path. Furthermore, the consumer's requests are mostly satisfied with nearby nodes. However, it leads to a lower stretch value.

Table 3: Effects of different parameters in path stretch

Parameters	Distance space	Request distance	Request space	Distance Request space
		Cache Size		
ALFA (α)	**0.04**	**0.04**	**0.04**	**0.04**
0.6	0.603	0.611	0.604	0.588
0.8	0.540	0.545	0.537	0.504
1	0.380	0.384	0.380	0.359
	0.08	**0.08**	**0.08**	**0.08**
0.6	0.571	0.579	0.570	0.553
0.8	0.518	0.527	0.519	0.496
1	0.366	0.373	0.365	0.358
	0.12	**0.12**	**0.12**	**0.12**
0.6	0.561	0.571	0.562	0.531
0.8	0.509	0.522	0.509	0.470
1	0.362	0.367	0.361	0.349
	0.16	**0.16**	**0.16**	**0.16**
0.6	0.546	0.556	0.546	0.509
0.8	0.471	0.485	0.473	0.449
1	0.361	0.366	0.364	0.352
	0.2	**0.2**	**0.2**	**0.2**
0.6	0.513	0.522	0.514	0.488
0.8	0.466	0.475	0.469	0.428
1	0.360	0.365	0.361	0.349

5.3 Latency

Latency refers to the request and content delivery delay reached by the consumer. Latency is measured in terms of time (millisecond) and is defined as follows:

$$Latency = Request\ Travel\ Delay + Content\ Travel\ Delay \tag{3}$$

Latency is a very important performance metric and it reflects the primary goal of in-network caching. Content redundancy, path stretch, and computationally expensive algorithms affect the latency. Low traffic and delay reduction are the basic objectives of in-network caching.

The COCP provides low latency because it reduces request and content travel delays. COCP caches the content at the most suitable location considering the consumer's request rate, the distance between consumer and publisher, and available cache space along the routing path. Furthermore, it has the potential to more content cached due to minimizing the content redundancy. Figs. 5a–5c show that the performance of the COCP is superior to other existing strategies in terms of latency with different popularity and cache sizes.

LCE represents a high latency because the contents are cached at all the routing nodes. It has no criteria to define the positioning of the cached content. CL4M finds the high betweenness centrality node along the path and then cached the content at that node. This technique failed to find out the node near to the user for caching the content. Therefore, it has very little chance to cache the content at the lower betweenness centrality nodes. ProbCache does not discriminate in position while the node's cache storage is overflowed. Therefore, there is a high possibility of content caching very far from the consumer which increases the content retrieval delay. LCD takes a long time for the content to reach the edge node because the content is cached on the adjacent node. Therefore, having the content cached close to the consumer depends on the cache hit. opt-Cache caches the required content to each node on the routing path. Furthermore, it associates a CGP value with every cached content calculated by distance and request rate. The content stays in the cache for a long time when the CGP value is high. So, it creates content redundancy and new content is less possible to be cached. In short, content diversity is low and most requests are adjusted by the publisher. The COCP uses distance, request rate, and available cache space to find the most suitable location for content caching. Moreover, it reduces redundancy and caches content near to the consumer. Therefore, the diversity of content in the network is high and maintains the balancing cache space along the routing path. However, more consumer's requests are satisfied in the network. COCP outperformed LCE, CL4M, ProbCache, LCD, and opt-Cache in terms of latency by 19.71%, 15.06%, 11.91%, 9.26%, and 5.54% respectively.

The network latency was also evaluated with various parameters. Tab. 4 shows how different parameters affect the network latency. When distance and request rate are used together, the latency is high because there are no criteria to evaluate the available space. So, the content becomes closer to the consumer but the edge nodes get exhausted from caching operations. Content has less cache due to longer distances from the consumer. As a result, there are long delays in retrieving content because the publisher mostly satisfies consumer requests. When distance and space parameters are used jointly, the latency is low because the distance parameter takes the content closer to the user and the space parameter assists the balancing between content dissemination according to the available cache space. Request rate with space combination gives better results because content caching starts from the edge node and move towards the publisher. However, distance/rate with space combinations has yielded better results compared to distance with the request rate combination. The last combination of parameters distance, request rate, and space, outperforms the previous three. This combination

provides the best place for content caching where retrieval delay of content is minimized. Moreover, it minimizes content pollution and the balance of cache space is used properly along the routing path.

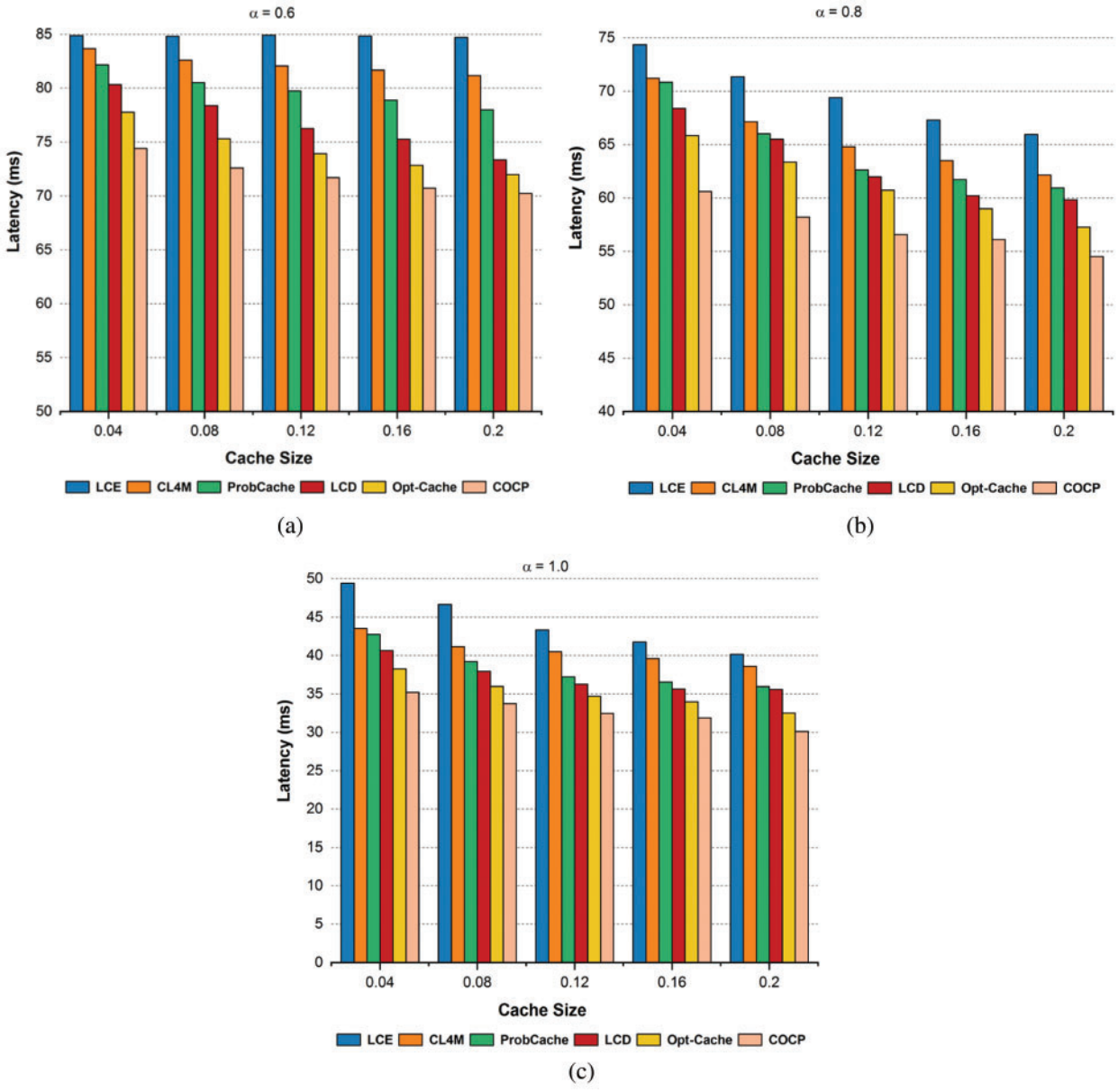

Figure 5: Latency with different cache sizes and α (a) Latency with different cache sizes and $\alpha = 0.6$ (b) Latency with different cache sizes and $\alpha = 0.8$ (c) Latency with different cache sizes and $\alpha = 1.0$

Table 4: Effects of different parameters in latency

Parameters	Distance space	Request distance	Request space	Distance request space
	Cache size			
ALFA (α)	**0.04**	**0.04**	**0.04**	**0.04**
0.6	80.18	81.32	80.25	74.42
0.8	65.05	65.75	64.98	60.60
1	37.23	38.06	37.14	35.22
	0.08	**0.08**	**0.08**	**0.08**
0.6	78.00	79.26	77.89	72.60
0.8	61.68	63.18	61.73	58.19
1	35.38	36.35	35.35	33.75
	0.12	**0.12**	**0.12**	**0.12**
0.6	76.29	77.99	76.51	71.70
0.8	60.46	62.12	60.30	56.57
1	34.75	35.08	34.66	32.45
	0.16	**0.16**	**0.16**	**0.16**
0.6	75.52	77.02	75.64	70.73
0.8	59.42	60.94	59.61	56.12
1	34.26	34.55	34.66	31.88
	0.2	**0.2**	**0.2**	**0.2**
0.6	75.28	76.37	75.31	70.24
0.8	58.74	59.71	59.17	54.52
1	34.12	34.57	33.99	30.10

5.4 Link Load

Link load reflects the number of bytes traversed on the link to obtain content from the source. It helps to measure the bandwidth utilization of the content in the network and is defined as follows:

$$Link\ Load = \frac{(request_{size} \times request_{link\_count}) + (content_{size} \times content_{link\_count})}{Duration} \tag{4}$$

$$Duration = Content\ Retrieval\ Time - Content\ Request\ Time \tag{5}$$

Figs. 6a–6c show that the performance of the COCP is better than other existing strategies in terms of link load. COCP manages the balancing between cache space along the routing path by considering the request rate, the distance between the publisher to the consumer, and available cache space. Therefore, the diversity of the content is high along the routing path because it minimizes the redundancy of content and avoids filling cache space quickly. Moreover, each content is cached at the most appropriate place where more interests are satisfied with the consumer's nearest node or from the routing path.

LCE represents a high link load due to the caching of content along with each routing node. The leads to wastage of storage and less diversity of content, which increases the link load. CL4M uses the concept of betweenness centrality for node selection. Furthermore, it does not use cache space properly as less betweenness centralized nodes have less content cached. However, the link load increased due to the high content retrieval delay. ProbCache fails to provide fairness in cache storage along the delivery path over different contents. It caches similar content to more than one node along the delivery path. Therefore, when it comes to downloading the content, it needs to go through several routers to find the right content. LCD takes a long time for the content to reach the edge node as it depends on the cache hit. So, the user request is completed far away from the edge node. LCD also increases the link load. opt-Cache is better than LCD in terms of link load because it considers the request rate and distance between the publisher to the consumer. It does not utilize cache space properly because it keeps unpopular content for a long time. Therefore, there is a very low possibility of new cached content. However, link load is high because the requested content is mostly accommodated far away from the consumer. COCP outperformed LCE, CL4M, ProbCache, LCD, and opt-Cache in terms of Link load by 38.14%, 33.85%, 30.75%, 27.03%, and 18.21% respectively.

Tab. 5 shows the effects of different parameters in terms of link load. The link load decreases when space, distance, and request rate are used jointly. The space parameter helped out in balancing of available cache space. Hence, there is a high possibility of storing more content along the routing path. When the distance and request rate is used jointly, it leads to a higher link load because most of the content is stored on the edge nodes. Therefore, the space of these nodes fills up faster and speeds up the process of eviction. So, content moves out quickly because of incoming content. This combination is not capable to handle cache space balancing along the routing path. As a result, there are fewer content stores and lower cache hits. However, the link load increased because most of the requests are satisfied by the publisher. The last combination is better than the previous combinations because caching is done at a suitable node and it also balanced the cache space along the path. We observed that whenever the space parameter is used with distance and rate, the load of the link decreases. This is because the distance and rate parameters will move the content closer to the consumer; however, when all three parameters are used together, it balances the cache space with the characteristic of these two parameters. The download path caches more content and minimizes the redundancy of content.

However, the consumer request is satisfied from the network with low path stretch which reduces the link load.

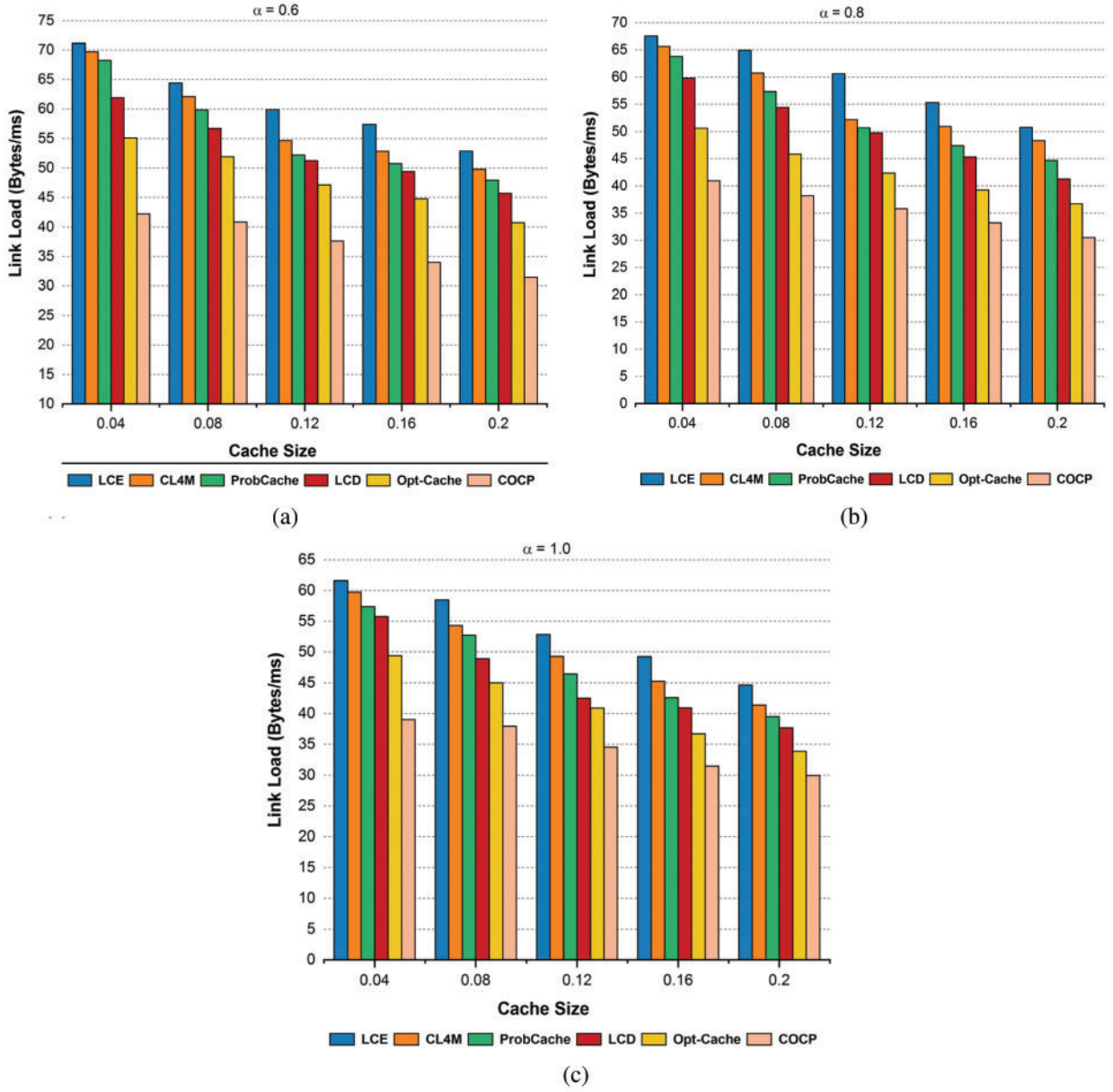

Figure 6: Link load with different cache sizes and α (a) Link Load with different cache sizes and $\alpha = 0.6$ (b) Link Load with different cache sizes and $\alpha = 0.8$ (c) Latency with different cache sizes and $\alpha = 1.0$

Table 5: Effects of different parameters in link load

Parameters	Distance space	Request distance	Request space	Distance request space
	Cache size			
ALFA (α)	**0.04**	**0.04**	**0.04**	**0.04**
0.6	48.48	50.04	48.95	42.24
0.8	45.74	46.07	45.39	40.95
1	42.56	43.46	42.17	39.02
	0.08	**0.08**	**0.08**	**0.08**
0.6	45.97	46.74	45.88	40.85
0.8	41.80	42.32	41.92	38.17
1	39.70	40.38	39.70	37.94
	0.12	**0.12**	**0.12**	**0.12**
0.6	40.88	43.23	41.08	37.62
0.8	38.19	40.71	38.08	35.86
1	35.86	37.12	35.10	34.53
	0.16	**0.16**	**0.16**	**0.16**
0.6	36.24	38.38	36.90	33.98
0.8	34.61	36.96	34.73	33.25
1	33.62	34.67	33.72	31.48
	0.2	**0.2**	**0.2**	**0.2**
0.6	34.07	35.79	34.95	31.47
0.8	32.72	33.65	32.76	30.53
1	31.58	32.77	31.04	29.95

6 Conclusion

In this study, a CCN-based content placement strategy named Coupling Parameters to Optimize Content Placement (COCP) is developed and evaluated to minimize the redundancy of contents along the downloading path. Existing in-network caching strategies have failed to used limited storage properly which leads to degrade cache hit ratio and network performance. COCP is a novel approach that utilizes the cache resources efficiently and enhances the performance of the network. We have performed extensive simulations under different constraints (cache sizes and content popularity). Conclusively, the results show that there is a decrease in latency and path stretch, a significant improvement in the cache hit ratio and the bandwidth is utilized properly over the CCN. The results obtained suggest that the proposed COCP strategy can handle the demands of future networks such as the Internet of Things (IoT) and vehicular networks.

Acknowledgement: The authors would like to acknowledge the support of Taif University Researchers Supporting Project number (TURSP-2020/292), Taif University, Taif, Saudi Arabia.

Funding Statement: This work was supported by Taif University Researchers Supporting Project Number (TURSP-2020/292), Taif University, Taif, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. Meng, M. A. Naeem, R. Ali and B. S. Kim, "EHCP: An efficient hybrid content placement strategy in named data network caching," *IEEE Access*, vol. 7, pp. 155601–155611, 2019.

[2] H. Noh and H. Song, "Progressive caching system for video streaming services over content centric network," *IEEE Access*, vol. 7, pp. 47079–47089, 2019.

[3] X. Zheng, G. Wang and Q. Zhao, "A cache placement strategy with energy consumption optimization in information-centric networking," *Future Internet*, vol. 11, no. 3, pp. 64–80, 2019.

[4] A. Ioannou and S. Weber, "A survey of caching policies and forwarding mechanisms in information-centric networking," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2847–2886, 2016.

[5] M. A. Naeem, S. A. Nor, S. Hassan and B. S. Kim, "Performances of probabilistic caching strategies in content centric networking," *IEEE Access*, vol. 6, pp. 58807–58825, 2018.

[6] M. A. Naeem, R. Ali, M. Alazab, M. Yhui and Y. Bin Zikria, "Enabling the content dissemination through caching in the state-of-the-art sustainable information and communication technologies," *Sustainable Cities and Society*, vol. 61, no. 102291, pp. 102291–102305, 2020.

[7] M. A. Naeem, M. A. U. Rehman, R. Ullah and B. S. Kim, "A comparative performance analysis of popularity-based caching strategies in named data networking," *IEEE Access*, vol. 8, pp. 50057–50077, 2020.

[8] F. Qazi, O. Khalid, R. N. Bin Rais and I. A. Khan, "Optimal content caching in content-centric networks," *Wireless Communications and Mobile Computing*, vol. 2019, pp. 1–15, 2019.

[9] H. Khattak, N. U. Amin, I. U. Din, I. Insafullah and J. Iqbal, "Leafpopdown: Leaf popular down caching strategy for information-centric networking," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 9, no. 2, pp. 148–151, 2018.

[10] M. A. Naeem, M. A. U. Rehman, R. Ullah and B. S. Kim, "A comparative performance analysis of popularity-based caching strategies in named data networking," *IEEE Access*, vol. 8, pp. 50057–50077, 2020.

[11] F. Khandaker, S. Oteafy, H. S. Hassanein and H. Farahat, "A functional taxonomy of caching schemes: Towards guided designs in information-centric networks," *Computer Networks*, vol. 165, no. 106937, pp. 106937, 2019.

[12] H. Wu, J. Li, J. Zhi, Y. Ren and L. Li, "Design and evaluation of probabilistic caching in information-centric networking," *IEEE Access*, vol. 6, pp. 32754–32768, 2018.
[13] B. Banerjee, A. Kulkarni and A. Seetharam, "Greedy caching: An optimized content placement strategy for information-centric networks," *Computer Networks*, vol. 140, pp. 78–91, 2018.
[14] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, K. Drira *et al.,* "Least fresh first cache replacement policy for NDN-based IoT networks," *Pervasive and Mobile Computing*, vol. 52, pp. 60–70, 2019.
[15] A. Seetharam, "On caching and routing in information-centric networks," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 204–209, 2017.
[16] Q. Wang, X. Zhu, Y. Ni, L. Gu, H. Zhao *et al.,* "A new content popularity probability based cache placement and replacement plan in CCN," in *Proc. ICCT*, Xi'an, China, pp. 1342–1347, 2019.
[17] W. K. Chai, D. He, I. Psaras and G. Pavlou, "Cache 'less for more' in information-centric networks," in *Proc. Int. Conf. on Research in Networking*, Berlin, Heidelberg, Germany, pp. 27–40, 2012.
[18] L. Saino, I. Psaras and G. Pavlou, "Icarus: A caching simulator for information centric networking (ICN)," in *Proc. ICST*, Lisbon, Portugal, vol. 7, pp. 66–75, 2014.
[19] Q. N. Nguyen, J. Liu, Z. Pan, I. Benkacem, T. Tsuda *et al.,* "PPCS: A progressive popularity-aware caching scheme for edge-based cache redundancy avoidance in information-centric networks," *Sensors*, vol. 19, no. 3, pp. 694, 2019.
[20] Y. Mordjana, M. R. Senouci and A. Mellouk, "Performance analysis of caching and forwarding strategies in content centric networking," in *Proc. GLOBECOM*, Singapore, pp. 1–6, 2017.
[21] M. A. Naeem, S. A. Nor, S. Hassan and B. -S. Kim, "Compound popular content caching strategy in named data networking," *Electronics*, vol. 8, no. 7, pp. 771–793, 2019.
[22] S. Arshad, B. Shahzaad, M. A. Azam, J. Loo, S. H. Ahmed *et al.,* "Hierarchical and flat-based hybrid naming scheme in content-centric networks of things," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1070–1080, 2018.
[23] A. Shinde and S. M. Chaware, "Content centric networks (CCN): A survey," in *Proc. I-SMAC*, Palladam, India, pp. 595–598, 2018.
[24] Y. Hayamizu, K. Hirata and M. Yamamoto, "CCAR: Caching and content-aware routing for content oriented networking," in *Proc. CQR*, Austin, TX, USA, pp. 1–6, 2018.
[25] A. Kalghoum, S. M. Gammar and L. A. Saidane, "Towards a novel cache replacement strategy for named data networking based on software defined networking," *Computers & Electrical Engineering*, vol. 66, pp. 98–113, 2018.
[26] K. N. Lal and A. Kumar, "A cache content replacement scheme for information centric network," *Procedia Computer Science*, vol. 89, pp. 73–81, 2016.