An Efficacious Content Caching and Eviction Priorities (CCEP) for In-network Caching High Performance in Information-centric Networking

Mohammad Alkhazaleh, S.A. Aljunid, and Naseer Sabri

***Abstract*—In-network caching is a popular information-centric networking (ICN) research topic due to the significant role in improving network performance to provide efficient data delivery. It is managed by a caching strategy that decides what, where, and when the content is cached. Accordingly, numerous caching strategies have been proposed to solve the drawbacks which existed in the default caching strategy to enhance ICN performance, although the majority of the proposed strategies were inadequate to produce efficient performance improvement. Meanwhile, popularity-based caching strategies are the most optimal approach, despite the disadvantages in calculating content popularity. Thus, this study presents an alternative caching strategy, namely Efficacious Content Caching and Eviction Priorities (CCEP), for ICN high performance. The CCEP could precisely calculate the content caching and eviction priorities based on the content popularity through significant factors. Simultaneously, the CCEP avoids the existing disadvantages in present caching strategies to obtain the highest performance in hop reduction, cache hit ratio, and the number of caching operations. The CCEP demonstrated significantly higher performance than the default caching strategy by elevating the hop reduction to 1667% and cache hit ratio by 1815%, while reducing the number of caching operations by 99%. The CCEP could also increase hop reduction by 53% and cache hit ratio by 17%, while reducing the number of caching operations to 21% compared to other improved caching strategies.**

***Index Terms*—Information-centric Networking, Caching Strategy, In-network Caching, Placement Mechanism, Replacement Mechanism**

I. Introduction

INFORMATION-centric networking (ICN) is an innovative infrastructure for the internet future [1] by caching the contents at different network routers to ensure content high availability and accessibility to users. Several architectures are encompassed in the ICN infrastructure based on the content name to retrieve the content from nearby routers rather than from the original server [2], [3]. Recently, Named Data Networking (NDN), one of the most prevalent ICN architectures [4], is an enhanced Content-centric Network (CCN) architecture version [5]. The ICN provides in-network caching features at every content router [6], [7] to allow efficient content retrieval and data delivery [8]. Correspondingly, the in-network cache requires an efficient caching strategy to manage different contents over ICN networks by deciding the content caching format, location, and time in the routers [7], [9]. Two mechanisms exist, namely placement and replacement. The placement mechanism is responsible for content caching, while the replacement mechanism serves to evict existing content when the cache is full to allow caching of new content arriving at the router [10], [11].

Leave Copy Everywhere (LCE) is the default ICN caching strategy [12], [3], which stores content at all content routers along the downloading path [5], [13]. Nonetheless, several problems exist, such as i) a redundancy increase, ii) a lower cache hit ratio, and iii) a larger number of caching operations, which would cause higher energy consumption [12], [14], [15]. Thus, different caching strategies have been proposed to resolve the disadvantages of the default caching strategy. Several proposed strategies are based on popularity, while other equivalents depend on distribution and probability respectively [16]. Existing proposed caching strategies retain certain disadvantages [17], [18], which render inefficient performance improvements. For instance, relevant featured content routers are unavailable to content caching in serving the users [17], [19], with cache redundancy and resource consumption remaining the major challenges [19]. Furthermore, certain proposed strategies do not contain a replacement mechanism or employ a mechanism only suitable for computers, such as Least Recently Used (LRU) and Least Frequently Used (LFU), which may be incompatible with ICN placement mechanisms [20]. Several strategies possess a high computational cost, which would be a significant implementation hurdle [12], whereas other strategies perform a large number of caching and eviction operations, which lead to high energy consumption, especially in wireless networks, due to the limited link capacity and battery energy [21].

Popularity-based caching strategies are the most optimal approach due to the avoidance of most weaknesses in the proposed caching strategies [22], [23]. Generally, popularity-based caching strategies mainly rely on two factors, namely i) the number of content requests in the content router and ii) the number of hops between the

Manuscript received April 10, 2022; revised December 09, 2022.

Dr. Mohammad Alkhazaleh is an Assistant Professor in Faculty of Information Technology, Isra University, Jordan. (e-mail: m.alkhazaleh@iu.edu.jo).

Dr. S. A. Aljunid is a Full Professor in School of Computer and Communication Engineering, University Malaysia Perlis, Malaysia. (e-mail: syedalwee@unimap.edu.my).

Dr. Naseer Sabri is an Assistant Professor in Technical Engineering College, Al-Farahidi University, Iraq. (e-mail: naseersabri@yahoo.com).

content router (CR) and the original server [14], [24]. Concurrently, the strategies calculate content popularity based on the number of content requests, although the number of requests is insufficient to decide whether the content is accurately popular at the CR. For example, the path length between the user and the server comprises four hops, and the number of requests for Content C at CR1, CR2, CR3, and CR4 is one, two, four, and eight respectively. Accordingly, popularity-based caching strategies would consider CR4 is with the highest content popularity. Nevertheless, if other CR4 cached contents possess a higher number of requests, such as nine, 10, and 12, Content C would not the most popular content. Simultaneously, when other CR3 cached contents possess a lower or equal number of requests, such as two, three, and four, Content C would be the most popular. Resultantly, solely focusing on the number of content requests is insufficient to decide the content popularity on a particular router. Comparison with the number of requests for other cached contents in the router is crucial to determine the popularity.

This study proposed and evaluated an alternative caching strategy, namely Efficacious Content Caching and Eviction Priorities (CCEP), for ICN high performance, to resolve all weaknesses in existing caching strategies. Moreover, the CCEP effectively employed influential factors to decide content popularity at a specific CR, while achieving the most critical performance metrics in terms of the cache hit ratio, hop reduction, and the number of caching operations [25]. Section 2 describes related ICN works, followed by Section 3 depicting the CCEP development with Section 4 delineating the CCEP performance evaluation.

II. Related Works

Numerous caching strategies have been proposed to resolve weaknesses in the default caching strategy, such as Leave Copy Down (LCD) [26], Move Copy Down (MCD) [26], Caching with Probability (Prob(p)) [26], Breadcrumbs [27], Probcache [28], WAVE [29], a chunk-caching position and searching scheme (CLS) [30], Most Popular Cache (MPC) [31], Edge Caching [32], Intra-Autonomous Systems Cache Cooperation (Intra-AS) [33], Centrality-based caching (CBC) [34], Two Layers Cooperative Caching (TLCC) [35], A Distributed MAx-Gain In-network Caching Strategy (MAGIC) [14], In-network Caching for ICN with Partitioning and Hash-routing (CPHR) [36], Cooperative In-network Caching (CINC) [37], Auction-based In-network Caching (BidCache) [38], Link Congestion and Lifetime-based In-networking Caching Schemes (LCLCS) [39], Adaptive Prioritized Probabilistic Caching Algorithm (APP) [40], and popularity and gain-based caching scheme (PGBCS) [24]. The existing caching strategies suffer from several disadvantages, which result in performance improvement inefficiency [17]. For enhanced clarity, the caching strategies are classified according to the respective disadvantages:

- The Intra-AS, TLCC, CPHR, CINC, and APP suffer from a lack of featured CRs, such as the edge router or the center, for content caching.
- The LCE, LCD, MCD, Breadcrumbs, Probcache, WAVE, CLS, Edge Caching, Intra-AS, CBC, TLCC, CPHR, and CINC suffer from featured content unavailability in providing in-network cache, such as popular content, to serve the users.
- The LCE, LCD, Breadcrumbs, and WAVE, encounter high cache redundancy and resource consumption by storing the same content repeatedly at each CR along the path between the user and server, which leads to high resource consumption.
- The MAGIC, BidCache, LCLCS, and PGBCS produce a high computational cost, which increases the implementation challenges.
- The LCE, LCD, MCD, Prob(p), Breadcrumbs, Probcache, WAVE, CLS, MPC, Edge Caching, Intra-AS, CBC, TLCC, CPHR, CINC, BidCache, and APP do not contain a replacement mechanism while employing computing replacement mechanisms, such as LRU and LFU, which may be incompatible with a proposed ICN placement mechanism. For example, a popularity-based caching strategy is proposed by applying LRU as a replacement mechanism. Although a popularity-based caching strategy attempts to cache popular content, the LRU mechanism may evict popular content to save unpopular content, which makes the replacement mechanism incomputable to the placement mechanism.
- The LCE, LCD, MCD, Prob(p), Breadcrumbs, WAVE, CLS, MPC, Edge Caching, Intra-AS, CBC, TLCC, CPHR, CINC, BidCache, LCLCS, APP, and PGBCS experience a massive number of caching and eviction operations, hence contributing to high energy consumption, especially in wireless networks, owing to the limited link capacity and battery energy.

After comparing the present caching strategies, MAGIC and Probcache provided the most optimal results [22], [23] according to the cache hit ratio, hop reduction, and the number of caching operations. MAGIC is a caching strategy which depends on popularity to calculate the local gain of content caching at each CR on the interest packet path. The CR with maximum gain would perform content caching [14]. Specifically, MAGIC calculates the place gain from the number of content requests in the CR and the number of hops between the CR and the original server. Subsequently, MAGIC calculates the replacement penalty for all cached contents to capture content with a minimum replacement penalty. The local gain is calculated by subtracting the minimum replacement penalty from the place gain. Resultantly, content popularity is calculated correctly to avoid existing aforementioned drawbacks. Nonetheless, calculating the replacement penalty for all cached contents would lead to a high computational cost, which increases the MAGIC application challenge [23].

Probcache is a probabilistic caching strategy, which is regarded as an optimal caching mechanism according to the number of caching operations. Two factors are integral to content caching, namely TimeIn, which is the maximum number of times afforded by the track to cache content, and the cache weight, which refers to the distance rate between the user to the CR and the user to the server [28]. Probcache calculates caching probability [ProbCache(x)] at each CR, with only a CR possessing a higher ProbCache(x) could cache the content. Nevertheless, several disadvantages remain in Probcache such as it has not replacement mechanism and featured content, which would cause performance improvement inefficiency.

III. The CCEP design

A. System Model

The system model consists of a set of CRs, with each CR equipped with a cache. The network was assumed as an undirected graph ($G =< CR, L >$), with a set of CRs ($CR= \{CR_1...CR_n\}$) and a set of links ($L= \{L_1...L_n\}$) between the CRs. The employed symbols are summarized in Table 1 to allow higher comprehension. Generally, in the NDN and CCN architectures, the user who requests content is required to follow two phases, namely i) the interest packet, which is the request packet sent from the user towards the content source, and ii) the data packet, which is the packet with content delivered to the user from the content source [41]. The NDN and CCN packets do not have a fixed header to provide the flexibility of developing the protocol continuously [42], [43].

TABLE I
Table of notation

Symbol	Meaning
CR	Content Router
c	Content
cc	Cached content
CP_c	The caching priority for content
MCP_c	The maximum caching priority for content
CR_{MCP}	The CR which has maximum caching priority for content
AvR	The average number of cached content requests at the CR
EP_c	The eviction priority for content
EP_{cc}	The eviction priority for cached content
MEP_{cc}	The maximum eviction priority for the cached content
d	The number of hops from the CR to the original server.
n	The number of cached contents in the CR
R_c	The number of requests for the content at the CR
R_{cc}	The number of requests for the cached content at the CR
C_{MEP}	The cached content with maximum eviction priority
CaS	The cache size
FS	The free space of the cache

B. The CCEP Caching Strategy

This study developed the CCEP caching strategy with two mechanisms, which were placement and replacement mechanisms, compatible with working principles to obtain the highest performance in the cache hit ratio, hop reduction, and the number of caching operations.

1) The Content Placement

The content placement is responsible for selecting a suitable content caching location. In the content placement stage, the CCEP caching strategy calculates every CR caching priority on the interest path, in which the CR with maximum caching priority would cache the content. A set of critical factors were applied to calculate the caching priority. The first factor was the number of content requests at the CR for high-popularity content caching. The second factor was the number of hops between the CR and the original server to cache content closer to the user. The third factor was the average number of cached content requests calculated by dividing two factors, namely the total numbers of cached content requests at CR and the number of cached content at CR, to obtain the average number of cached content requests and ensure that the content was popular at the CR.

Solely referring to the number of content requests was inadequate to decide content popularity accurately at the CR, owing to the numbers of other content requests would affect content popularity. Therefore, the average number of cached content requests was employed to calculate the content popularity. As the average number of cached content requests is inversely proportional to content popularity, the average number of cached content requests at the CR was first determined through Equation 1 to define the caching priority:

$$AvR = \frac{\sum_{cc=1}^{n} R_{cc}}{n} \quad (1)$$

Where R_{cc} is the number of cached content requests and n is the number of cached content in the CR. Subsequently, the caching priority could be calculated via Equation 2:

$$CP_c = \frac{R_c}{AvR} \times d \quad (2)$$

Where R_c is the number of new content requests at the CR, d is the number of hops between the CR and the original server, and AvR is the average number of cached content requests at the CR. From the previous equation, the CCEP strategy divided R_c by AvR, to ensure that the content was popular at the CR.

2) The Content Replacement

The content replacement is responsible for evicting content when the cache is full to allow caching of new content arriving at the router. In the content replacement stage, the CCEP strategy calculates eviction priority for cached content, wherein the content with maximum eviction priority would be evicted to cache the new content. Two critical factors were applied to calculate eviction priority. The first factor was the number of interfaces which requested content to evict unpopular content while caching popular content, as popular content caching would increase the cache hit ratio. The second factor is the number of hops between the CR and the original server to carry far content closer to users in shortening the retrieval time. The eviction priority could be calculated using Equation 3:

$$EP_{cc} = \frac{d}{R_{cc}} \quad (3)$$

Where R_{cc} is the number of cached content requests and d is the number of hops between the CR and the original server. Hence, an alternative caching strategy, namely the CCEP, was proposed for in-network caching high performance.

3) Additional Interest and Data Packet Header Fields

Two fields were added to the interest and data packet headers to exchange information in managing the internal contents of the in-network cache. The fields also assisted the CCEP strategy in content caching at a single router along the download path. Meanwhile, the selected CR was the most optimal content-caching router due to the user proximity and ability to support future requests, which would improve the cache performance. The update was not arbitrary as the NDN and CCN packet formats did not contain a fixed packet header to provide flexibility in developing the protocol continuously [42].

C. CCEP Algorithms

Two algorithms were employed in the CCEP caching strategy, namely, i) the interest packet for placement and ii) the data packet for replacement.

1) CCEP Algorithm (Interest Packet)

The interest packet is created in the CCEP when a user requests the content before two fields are added in the header to carry the CR information. The first field is the maximum caching priority for content (MCP_c), whereas the second field is the router with maximum caching priority (CR_{MCP}).

As portrayed in Figure 1, matching content is searched in the content store (CS) when an interest packet arrives at a CR, in which the content is sent to the user when the content is available. Conversely, the interest packet record is checked in the Pending Interest Table (PIT), which is the default ICN data structure. The incoming interface is added to the existing interface list of interfaces when the entry is in the PIT. Otherwise, an additional entry is added in the PIT. Subsequently, the number of interfaces which requested the content is calculated, before calculating the AvR through Equation 1 and the CP_c via Equation 2. If the CP_c value is larger than the MCP_c value, the MCP_c value is replaced with the CP_c value, while the CR_{MCP} value is substituted by the current CR value of the current CR. Otherwise, the values remain the same. Ensuingly, the interest packet is forwarded to the following CR, where the CCEP strategy repeats the previous steps. Eventually, the CCEP strategy captures the CR maximum priority (CR_{MCP}) when the cache hits.

2) CCEP Algorithm (Data Packet)

At the interest packet phase end with the capturing of the CR_{MCP} value, the CCEP strategy adds the CR_{MCP} value to the data packet header to locate the current CR.

Figure 2 illustrates that the CCEP strategy would check whether the CR cache possesses FS when the data packet is not delivered to the user and arrives at the CR. If FS is available, the content would be cached before the data packet is forwarded to the following CR. Contrarily, the strategy would check that the current CR is CR_{MCP}, if no, the data packet is forwarded to the following CR. Contrarily, the strategy calculate the EP_c value. Subsequently, the EP_{cc} is

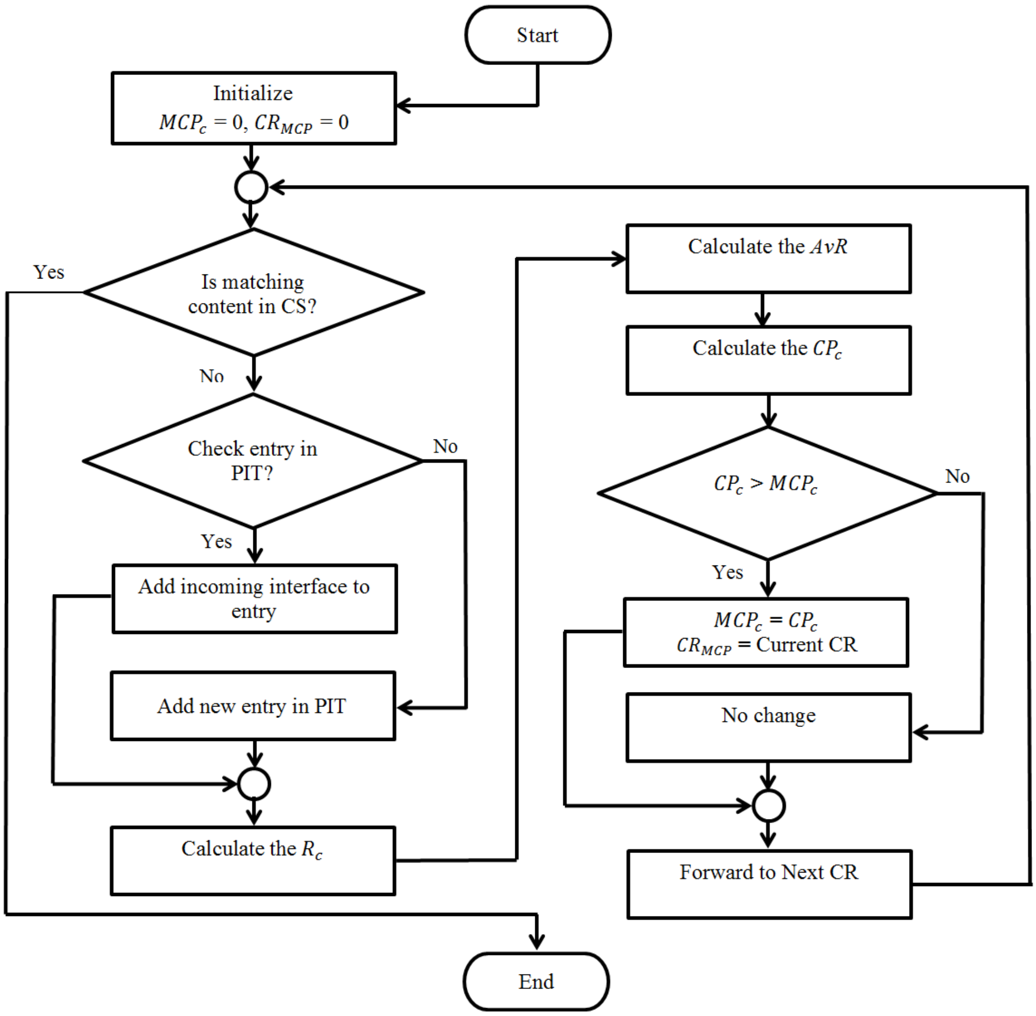

Fig. 1 CCEP Algorithm (Interest Packet)

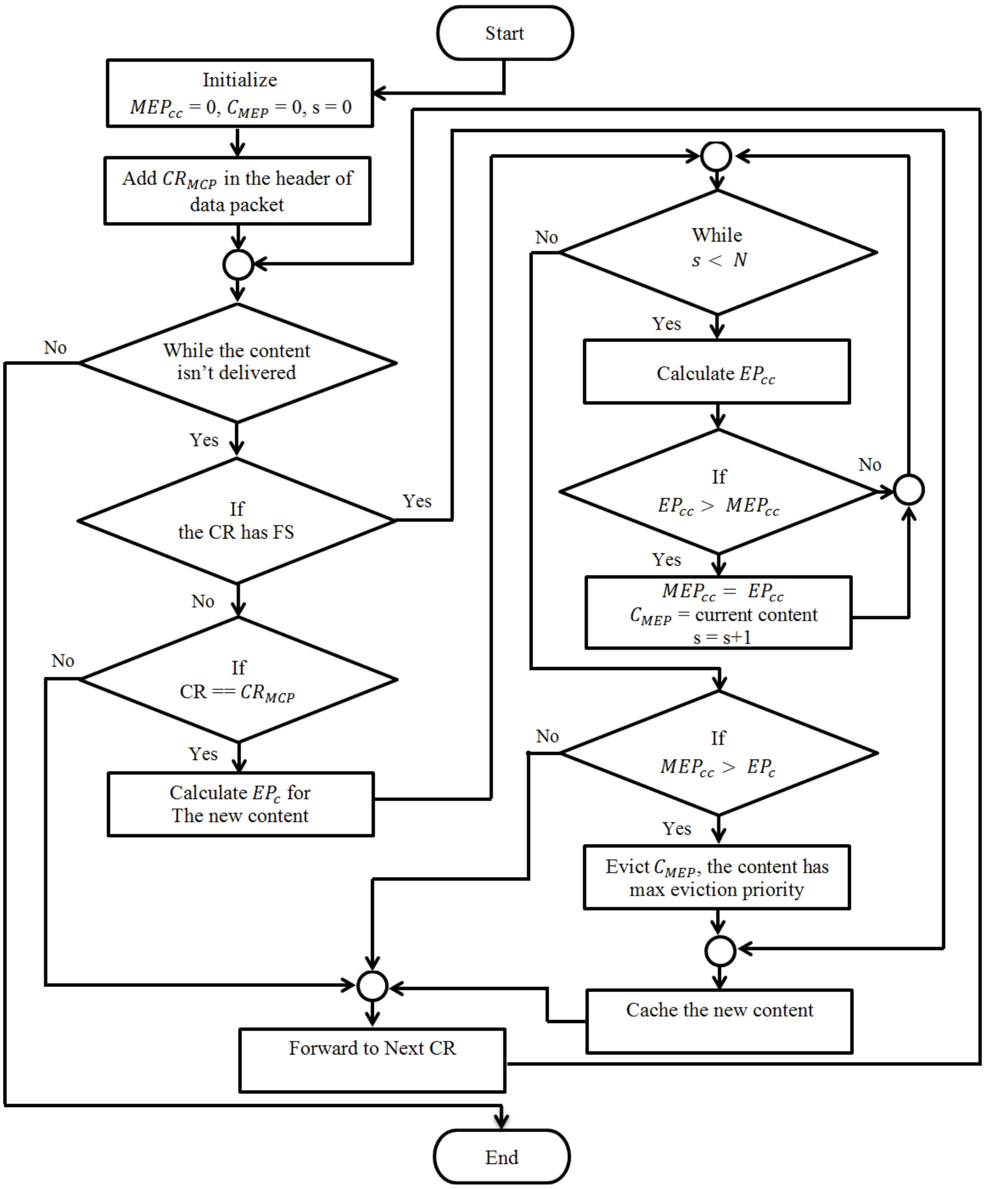

Fig. 2. CCEP Algorithm (Data Packet).

reckoned to acquire the MEP_{cc} in determining whether the MEP_{cc} is larger than the EP_c. A larger MEP_{cc} value would evict the C_{MEP} before caching new content and forwarding the data packet to the following CR. Conversely, the data packet is forwarded to the following CR without caching content to reduce caching operations.

D. A CCEP Design Summary

Summarily, the CCEP caching strategy calculates content popularity more accurately than the existing caching strategies, while avoiding all present disadvantages to obtain in-network caching high performance as follows:

- The CCEP caches content at a featured CR, which has high content popularity and is closer to the user.
- The CCEP caches the most popular content.
- The CCEP caches content at one CR along the downloading path, which has the highest content caching priority to reduce content redundancy.
- The CCEP strategy possesses a low computational cost as the strategy efficiently calculates the suitable content caching location.
- The CCEP employs a replacement mechanism computable with the placement mechanism while avoiding the replacement mechanisms applied in computers.
- The CCEP reduces the number of caching operations by caching only popular content at a single router along the downloading path to allow longer caching time.

IV. CCEP Performance Evaluation

A. Simulation Scenarios

The SocialCCNSim [44] simulator was adopted to evaluate the CCEP performance due to the availability as open-source software with the most caching strategies for comparison. The CCEP caching strategy was compared with three caching strategies (LCE, Probcache, and MAGIC) according to the LRU cache replacement policy, which was perceived as the most appropriate replacement policy [45].

To ensure a fair evaluation, CCEP, LCE, Probcache, and MAGIC were simulated in the same parameters and topologies (GEANT and DTelecom), as depicted in Table 2. Simultaneously, different cache sizes (2 GB – 8 GB) and different popularity distributions (Zipf: α = 0.65 – 2.0) were selected to ensure that the strategy was valid for all cases. Meanwhile, the numbers of contents, users, and Zipf distributions were selected as regular traffic indicators based on SONETOR, which is a set of utilities that generates synthetic social network traces [46].

TABLE II
EXPERIMENTAL PARAMETERS

Parameter	Values
Number of Users	3980
Number of Contents	100000
Content size	10 MB
Cache size	2 GB - 8 GB (200 - 800 contents)
Access pattern	Zipf: α = 0.65 – 2.0
No. of Simulation	8 Runs
Topology	GEANT, DTelecom

B. Performance Metrics

The performance of the caching strategies was evaluated from three aspects:

Hop reduction ratio. The metric would indicate the saved bandwidth consumption [47], wherein the hop reduction ratio was calculated through Equation 4.

$$\beta = \frac{\sum h}{\sum H} \tag{4}$$

Where h is the hop counts from the CR (where a cache hit occurs) to the original server, and H is the hop count from the client to the original server [48].

Cache hit ratio. The metric is the number of interest messages answered from the cache rather than from the content source. The ratio is represented by a fraction recognized as the cache-hit ratio in Equation 5 [19]. Occasionally, the metric is regarded as cache-hit probability or cache-hit rate.

$$Cache\_Hit = \frac{\sum_{n=1}^{n} hits_i}{\sum_{n=1}^{n} hits_i + \sum_{n=1}^{n} miss_i} \tag{5}$$

Where $hits_i$ is the number of interest messages answered by the cache of the CR i, $miss_i$ is the number of interest messages answered by the content source, and n is the number of CRs in the network topology [49].

The number of caching operations. The metric is the number of cache input and output (I/O) operations at all CRs [14], which was calculated via Equation 6:

$$\gamma = \sum_{n=1}^{n} O_i \tag{6}$$

Where O_i is the number of cache I/O operations at CR i and n is the CR number.

C. Network Topologies

Numerous topologies were employed to simulate the existing caching strategies. The topologies are classified into realistic topologies and synthetic topologies. To ensure satisfactory assessment, two realistic topologies (GEANT and DTelecom) were selected for simulation to ensure the CCEP strategy applicability in different networks and ascertain the CCEP strategy validation by properly evaluating the performance. The topologies are described below:

The GEANT topology is a simple-realistic topology consisting of 22 CRs and 37 links, as portrayed in Figure 3. The topology is sketched from the GEANT network which interconnects the European National Research and Education Networks.

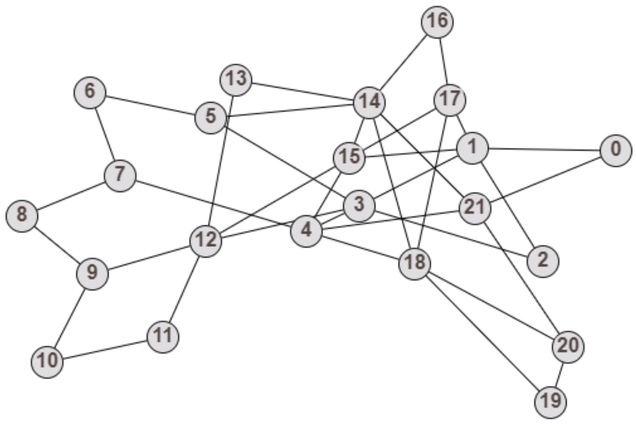

Fig. 3. GEANT Topology.

The DTelecom topology is a complex-realistic topology comprising 68 CRs and 350 links, as illustrated in Figure 4. The topology is developed from the Deutsche Telekom network, which is one of the largest European telecommunication providers.

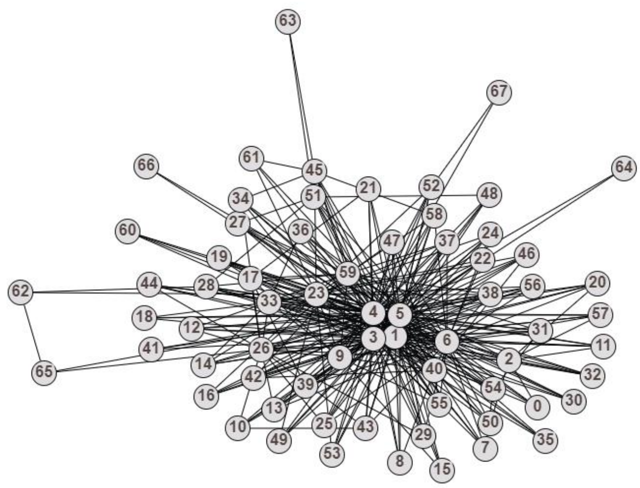

Fig. 4. DTelecom Topology.

D. Simulation Results

The CCEP strategy simulation was performed using two topologies, GEANT and DTelecom. Therefore, the results were described in two parts as follows:

1) The Results through the GEANT Topology

The CCEP results via the GEANT topology were discussed and compared with those of LCE, Probcache, and MAGIC in terms of the hop reduction ratio, cache hit ratio, and the number of caching operations.

a) The Hop Reduction

Figure 5 depicts the hop reduction ratio for the strategies, with different Zipf distributions (α = 0.65, 1.10, 1.50, and 2.00) and cache sizes (two, four, six, and eight GB). Overall, the CCEP achieved the highest performance results in terms of the hop reduction ratio in all test cases. Concurrently, the hop reduction rate of all caching strategies elevated with the increases in cache size and alpha. Similarly, the CCEP improvement rate constantly maintained the maximum performance improvement degree compared to other caching strategies when the cache size was small (two GB). Conversely, the improvement rate decreased when the cache size increased.

When the alpha value was small (0.65), a big difference was recorded when comparing the CCEP performance with LCE and Probcache. The maximum CCEP improvement rate was at (0.0076-0.0005)/0.0005 = 1420% with Probcache and (0.0076-0.00043)/0.00043 = 1667% with LCE respectively. Contrarily, the CCEP improvement rate decreased when increasing the alpha value to (0.91036-0.86644)/0.86644 = 5.07% with Probcache and to (0.91036-0.86368)/0.8637 = 5.4% with LCE respectively, with the alpha value being 2.0 and the cache size being eight GB. The maximum CCEP performance improvement rate was (0.50488-0.32943)/0.32943 = 53.26% when compared with the MAGIC. Nonetheless, the CCEP performance decreased to attain a similar rate as that of MAGIC when the alpha value was 2.0 with a cache size of eight GB.

b) The Cache Hit

Figure 6 highlights the cache hit ratio between the four caching strategies, with different Zipf distributions (α = 0.65, 1.10, 1.50, and 2.0) and different cache sizes (two, four, six, and eight GB). The CCEP achieved the highest cache hit ratio in all test cases. Simultaneously, the cache hit ratio value of each caching strategy increased when the alpha value and the cache size increased. Nevertheless, when the alpha value was small (0.65), a huge difference in performance improvement was recorded between CCEP and LCE and between CCEP and Probcache. The maximum CCEP improvement rate was at (0.1226-0.0005)/0.0005 = 2352% with Probcache and (0.1226-0.00064)/0.00064 =

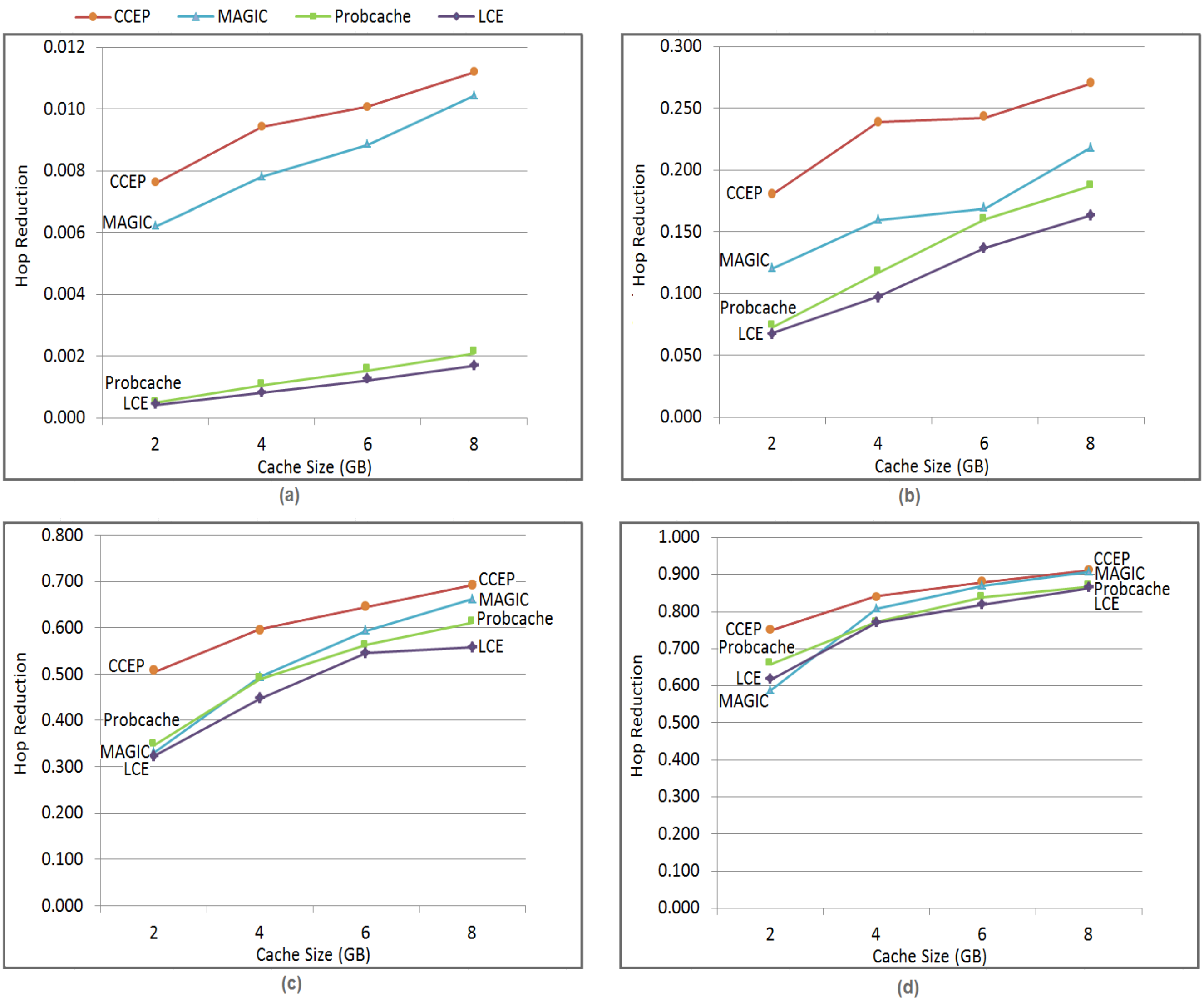

Fig. 5 The hop reduction ratio results via GEANT topology with different Zipf distributions, a = 0.65, b = 1.10, c = 1.50, and d = 2.00.

1815% with LCE respectively. The CCEP improvement rate decreased by the alpha value increase of (0.80358-0.72226)/0.72226 = 11.26% with Probcache and (0.80358-0.7041)/0.7041 = 14.13% with LCE, respectively, when the alpha value was 2.0 and the cache size was eight GB.

The CCEP performance was almost similar to MAGIC when the alpha value was between 0.65 and 1.1. Contrarily, when the alpha value was between 1.5 and 2.0, the maximum performance improvement rate was (0.33336-0.3051)/0.3051 = 9.26% and (0.55779-0.475)/0.475 = 17.43% respectively, when the alpha value was between 1.5 and 2.0, and the cache size was two GB. Meanwhile, the CCEP improvement rate was decreased to (0.50325-0.48673)/0.48673 = 3.39% and (0.80358-0.79413)/0.79413 = 1.19% respectively, when cache size was eight GB. Specifically, when the alpha value was between 1.5 and 2.0, the CCEP performance improvement decreased by the cache size increase, which suggested the highest performance at two GB cache size.

c) The Caching Operations

Figure 7 delineates the number of caching operations for the four strategies through the GEANT topology, with different Zipf distributions (α = 0.65, 1.10, 1.50, and 2.0) and different cache sizes (two, four, six, and eight GB). Accordingly, CCEP achieved the most optimal result in all test cases. Moreover, a significant performance difference was recorded between the CCEP and MAGIC when compared to Probcache and LCE. For example, when the alpha value was 0.65 and the cache size was two GB, the number of CCEP caching operations and that of MAGIC was 236 and 254 respectively. Nonetheless, the LCE and Probcache operation numbers were 1,296,928 and 710,270 respectively. Thus, the CCEP and LCE graphs were enlarged to provide a clearer visual on the right side of Figure 6.

The number of caching operations for all caching strategies decreased with the alpha value increase. Similarly, the LCE and Probcache numbers of caching operations decreased when the cache size increased. Contrarily, the CCEP and MAGIC numbers of caching operations increased when the cache size was expanded. The CCEP performance improvement rate was higher than 99% in all test cases compared to LCE and Probcache.

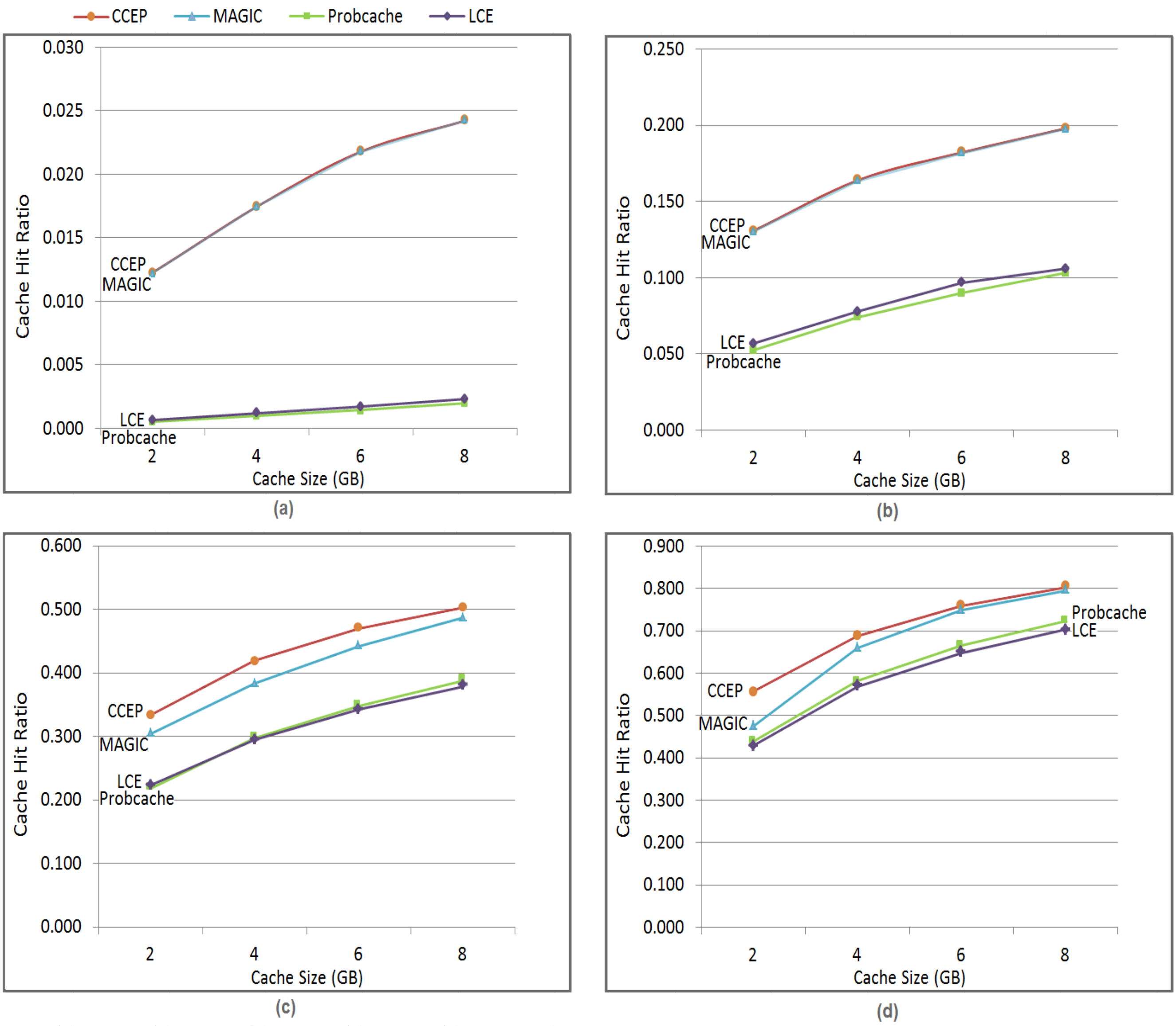

Fig. 6. The cache hit ratio results via the GEANT topology with different Zipf distributions, a = 0.65, b = 1.10, c = 1.50, and d = 2.00.

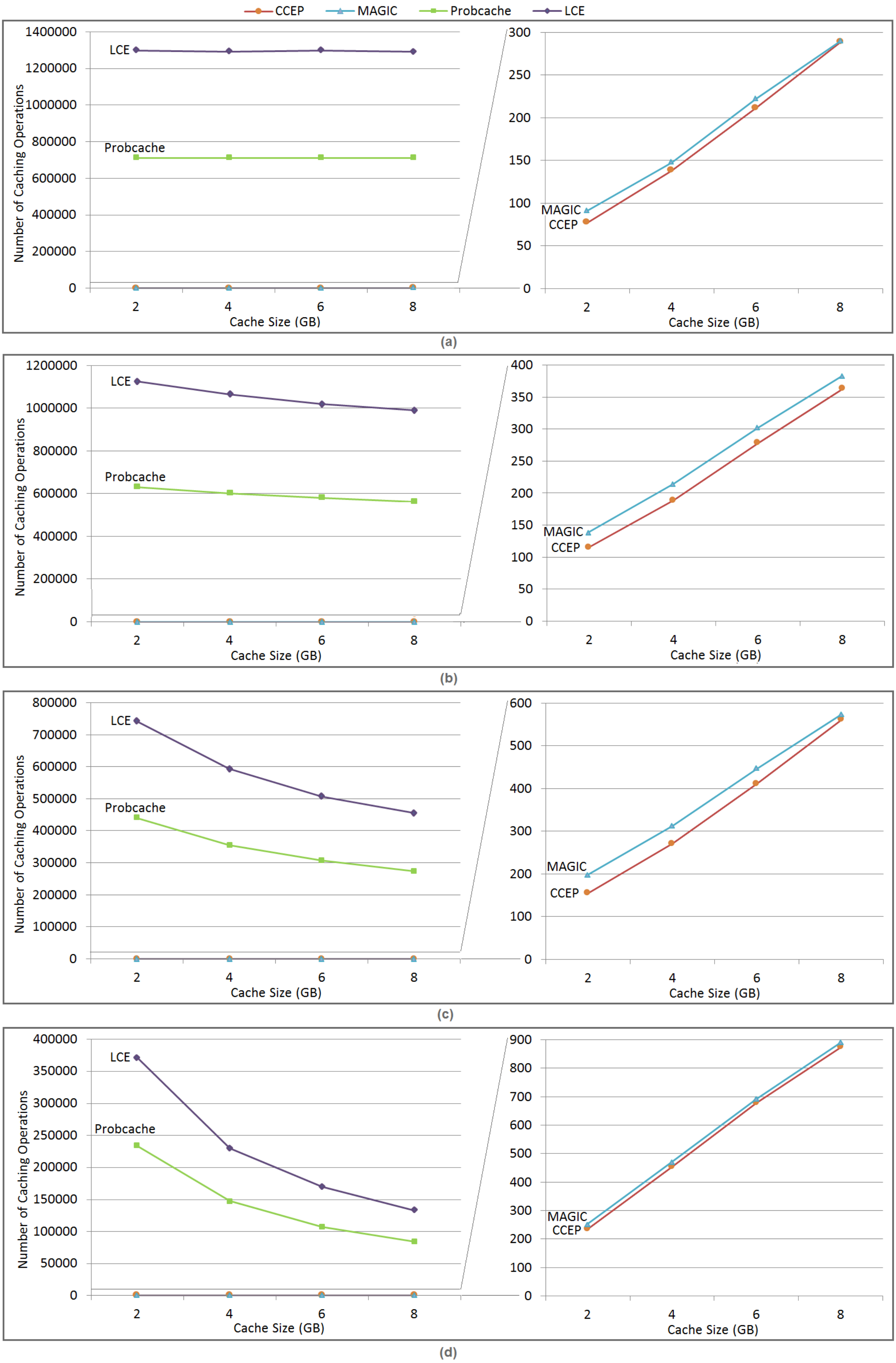

Fig. 7. The findings of the number of caching operations through the GEANT topology with different Zipf distributions, a = 0.65, b = 1.10, c = 1.50, and d = 2.00.

The maximum CCEP performance improvement was -(155-198)/198 = 21.65% compared to MAGIC, although the rate decreased to be similar to MAGIC when the alpha value was 2.0 and the cache size was eight GB. Nevertheless, when the cache size was small (two GB), the CCEP performance improvement achieved the maximum rate compared to other caching strategies, which would also decrease when the cache size increased.

2) The Results through the DTelecom topology

The CCEP results through the DTelecom topology were compared with LCE, Probcache, and MAGIC in terms of the hop reduction ratio, cache hit ratio, and the number of caching operations.

a) The Hop Reduction

Figure 8 depicts the hop reduction ratio between the strategies using DTelecom topology, with different Zipf distributions (α = 0.65, 1.10, 1.50, and 2.0) and different cache sizes (two, four, six, and eight GB). The CCEP achieved the highest hop reduction ratio in all test cases, while the hop reduction ratio for all caching strategies increased with the increase in the cache size and the alpha value. Nevertheless, the CCEP performance improvement was constantly at the maximum rate compared to other caching strategies when the cache size was small (two GB), whereas the improvement rate decreased when the cache size was enlarged.

A significant performance difference was recorded between CCEP and LCE and between CCEP and Probcache when the alpha value was small (0.65). The maximum CCEP improvement rate was at (0.00696-0.00053)/0.00053 = 1213% with Probcache and (0.00696-0.00043)/0.00043 = 1518% with LCE respectively. Meanwhile, the improvement rate decreased with the alpha value being increased to (0.90195-0.86220)/0.86220 = 4.61% with Probcache and to (0.90195-0.86179)/0.86179 = 4.66% with LCE, when the alpha value was 2.0 and the cache size was eight GB. The maximum CCEP performance improvement rate was (0.15269-0.11241)/0.11241 = 35.83% compared to MAGIC, which would be decreased to be similar to MAGIC with the cache size being eight GB.

b) The Cache Hit Ratio

Figure 9 demonstrates the cache hit ratio for the strategies via the DTelecom topology, with different Zipf distributions

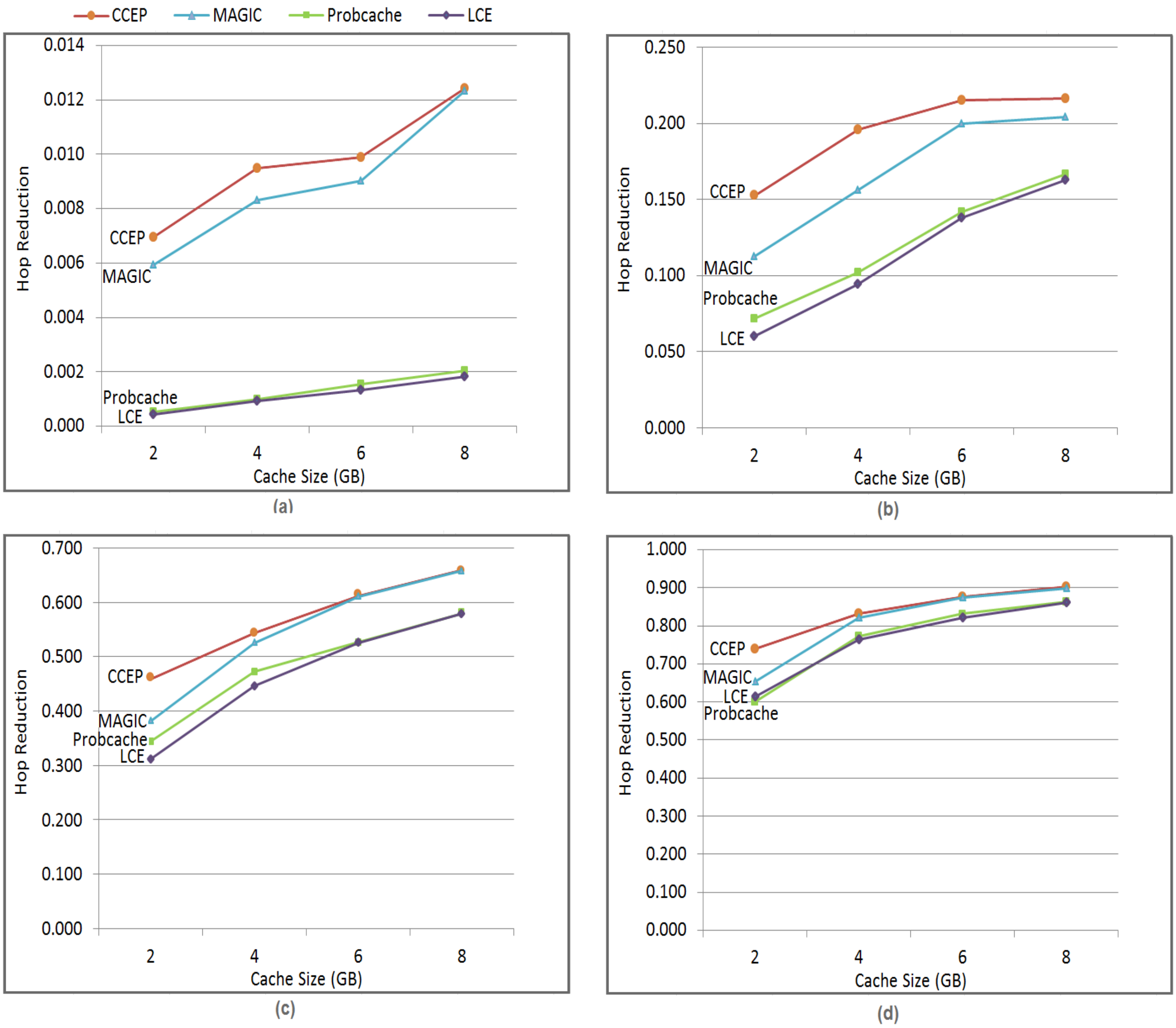

Fig. 8. The hop reduction ratio results via the DTelecom topology with different Zipf distributions, a = 0.65, b = 1.10, c = 1.50, and d = 2.00.

(α = 0.65, 1.10, 1.50, and 2.0) and different cache size values (two, four, six, and eight GB). CCEP achieved the highest cache hit ratio in all test cases, with the cache hit ratio value of all caching strategies increasing when the cache size and alpha value increase.

A huge performance difference was recorded between CCEP and LCE and between CCEP and Probcache when the alpha value was small (0.65). The maximum CCEP improvement rate was at (0.02663-0.00199)/0.00199 = 1238% with Probcache and (0.02663-0.00293)/0.00293 = 808% with LCE respectively. Nevertheless, the improvement rate decreased when the alpha value was increased to (0.84101-0.77353)/0.77353 = 8.72% with Probcache and (0.84101-0.77411)/0.77411 = 8.64% with LCE respectively when the alpha value was 2.0 and the cache size was eight GB.

The CCEP performance was almost similar to the MAGIC when the alpha value was between 0.65 and 1.1. Contrarily, when the alpha value was between 1.5 and 2.0, the maximum performance improvement was (0.43924-0.41740)/0.41740 = 5.23% and (0.63600-0.58843)/0.58843 = 8.08% respectively, when the alpha value was between 1.5 and 2.0 and the cache size was two GB. The CCEP improvement rate would also be decreased to be similar to MAGIC when the cache size was eight GB.

c) The Caching Operations

Figure 10 illustrates the number of caching operations for the strategies via the DTelecom topology, with different Zipf distributions (α = 0.65, 1.10, 1.50, and 2.0) and different cache sizes (two, four, six, and eight GB). CCEP achieved the most optimal results in the number of caching operations in all test cases, in which a high difference was recorded between CCEP and MAGIC compared to LCE and Probcache. For example, when the alpha value was 0.65 and the cache size was two GB, the CCEP and MAGIC numbers of caching operations were 693 and 747 respectively, while those of LCE and Probcache were 1,062,594 and 663,626 respectively. The CCEP and MAGIC graphs were expanded to provide an apparent visual on the right side of Figure 6.

The number of caching operations for all caching strategies decreased with the increase in the alpha value, while only the LCE and Probcache would exhibit a decrease in the number of caching operations when the cache size increased.

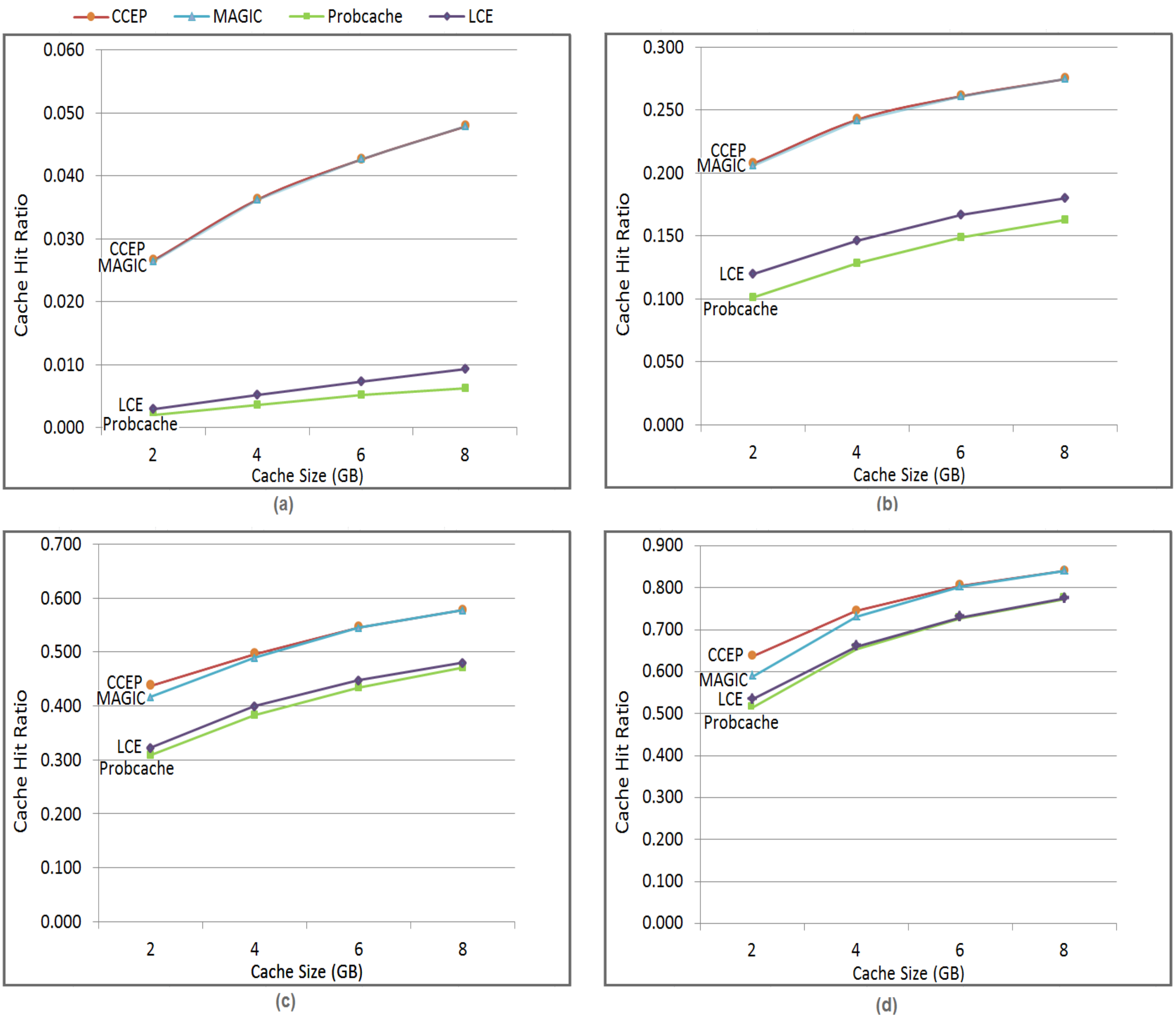

Fig. 9. The cache hit ratio findings through the DTelecom topology with different Zipf distributions, a = 0.65, b = 1.10, c = 1.50, and d = 2.00.

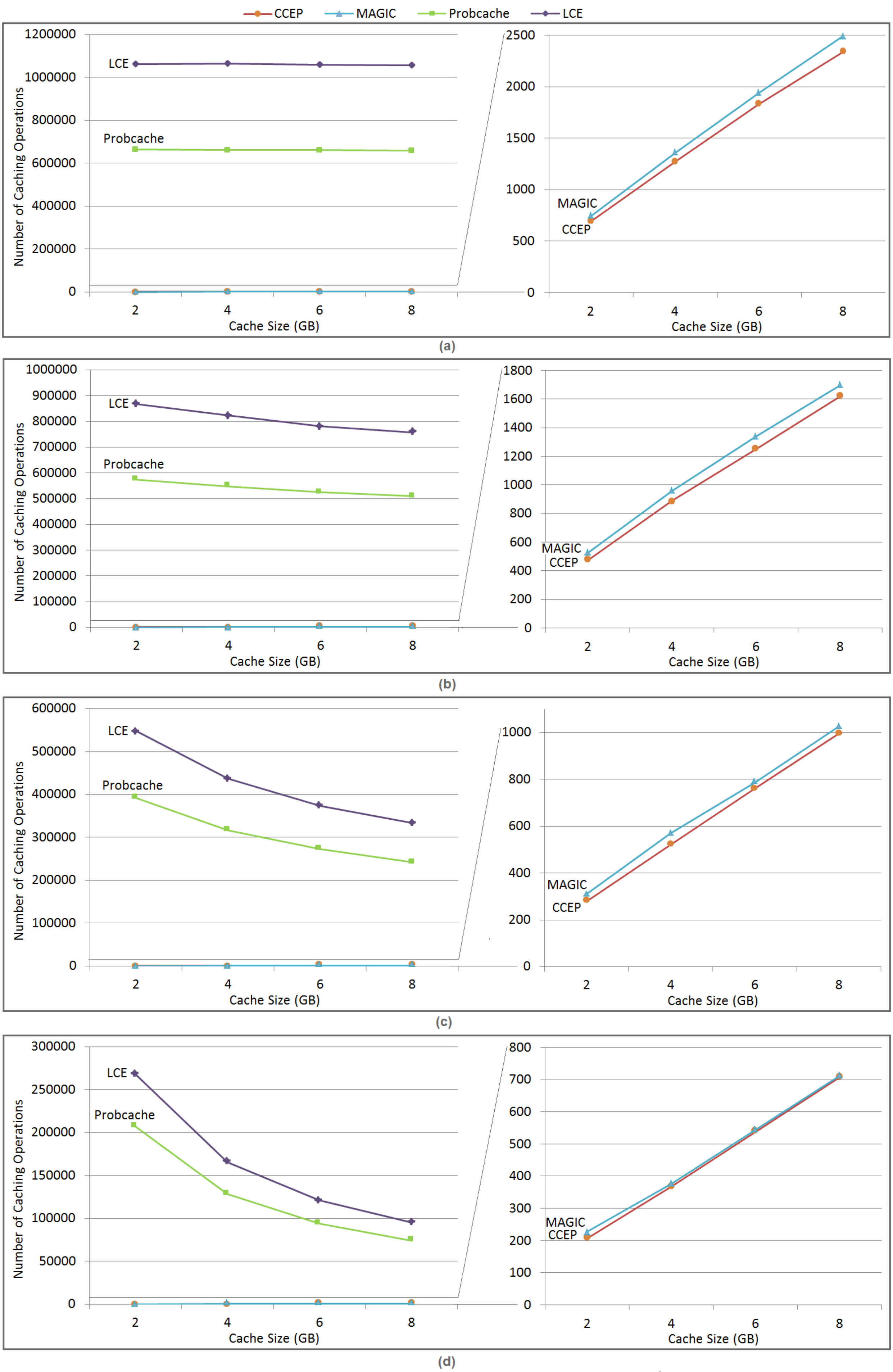

Fig. 10. The results of the number of caching operations through the DTelecom topology with different Zipf distributions, a = 0.65, b = 1.10, c = 1.50, and d = 2.00.

Conversely, the CCEP and MAGIC numbers of caching operations increased when the cache size was enlarged. Nevertheless, the CCEP performance improvement accomplished the maximum rate compared to other caching strategies when the cache size was small (two GB), despite the improvement rate decreasing with the increase in the cache size. Hence, the CCEP performance improvement rate was higher than 99% in all test cases compared to LCE and Probcache, with the maximum improvement rate at -(475-526)/526 = 9.79% compared to MAGIC. Nonetheless, the CCEP improvement rate decreased to be similar to MAGIC when the alpha value was 2.0 and the cache size was eight GB.

3) The Results between the GEANT Topology and the DTelecom Topology

The hop reduction results through the GEANT topology were discovered to be slightly higher than via the DTelecom topology. Conversely, the cache hit ratio results through the GEANT topology were slightly lower than via the DTelecom topology. In addition, the number of caching operations through the GEANT topology was significantly lower than via the DTelecom topology. The difference was owing to the number of CRs in the GEANT topology being lower than in the DTelecom topology. Meanwhile, the CCEP performance improvement rate through the GEANT topology was higher than via the DTelecom topology in all performance metrics.

V. Conclusion

The present study proposed the CCEP to enhance the in-network caching performance in terms of the hop reduction ratio, cache hit ratio, and the number of caching operations. Particularly, all weaknesses that existed in previous caching strategies were avoided, while employing significant factors to calculate content popularity accurately and efficiently to acquire the most optimal content caching location. The simulation results revealed that the CCEP achieved the highest performance in terms of the hop reduction ratio, cache hit ratio, and the number of caching operations compared to other caching strategies. Furthermore, the CCEP was compared with the default caching strategy, wherein the number of caching operations was reduced by 99% while increasing the hop reduction by 1667% and the cache hit ratio by 1815%. Simultaneously, the CCEP was compared with the most optimal caching strategy, namely MAGIC. Specifically, the CCEP ably reduced the number of caching operations by 21%, while increasing the hop reduction ratio by 53% and the cache hit ratio by 17% compared to MAGIC. Summarily, the CCEP is considered the most pertinent caching strategy for the internet future.

References

[1] M. Amadeo, C. Campolo, and A. Molinaro, "Information-Centric Networking for Connected Vehicles : A Survey and Future Perspectives," *IEEE Commun*. Mag., vol. 54, no. 2, pp. 98–104, 2016.

[2] L. Zhu et al., "T-CAM: Time-based content access control mechanism for ICN subscription systems," *Futur. Gener. Comput. Syst.*, vol. 106, pp. 607–621, 2020.

[3] G. Zhang, Y. Li, and T. Lin, "Caching in information-centric networking: A survey," *Comput. Networks*, vol. 57, no. 16, pp. 3128–3141, 2013.

[4] Hussaini, M., Nor, S. A., Ahmad, A., Mustapha, R., & Abdulateef, A. F, "A conceptual model of producer mobility support for named data networking using design research methodology," *IAENG International Journal of Computer Science,* vol. 46, no. 4, pp. 552–563, 2019.

[5] A. Aboodi, T. C. Wan, and G. C. Sodhy, "Survey on the Incorporation of NDN/CCN in IoT," *IEEE Access*, vol. 7, pp. 71827–71858, 2019.

[6] S. Lo, J. Hu, and V. A. Kshirsagar, "Neighborhood-Based In-Network Caching for Information-Centric Networks," *Int. J. Commun. Netw. Syst. Sci.*, vol. 10, no. 8, pp. 76–87, 2017.

[7] C. Bernardini, T. Silverston, and A. Vasilakos, "Caching Strategies for Information-Centric Networking: Opportunities and Challenges," pp. 1–14, Jun. 2016. [Online] Available: https://arxiv.org/abs/1606.07630.

[8] Anjali, "In-network Caching Scheme in Information Centric Networking," *Int. J. New Innov. Eng. Technol*., vol. 5, no. 1, pp. 113–120, 2016.

[9] M. Alkhazaleh, S. A. Aljunid, and N. Sabri, "a Comprehensive Survey of Information-Centric Network : Content Caching Strategies," *J. Crit. Rev.*, vol. 7, no. 04, pp. 2272–2287, 2020.

[10] Y. Sun et al., "Trace-driven analysis of ICN caching algorithms on video-on-demand workloads," *in Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, 2014, pp. 363–375.

[11] C. Li and K. Okamura, "Cluster-based In-networking Caching for Content-Centric Networking," *Int. J. Comput. Sci. Netw. Secur.*, vol. 14, no. 11, pp. 1–9, 2014.

[12] F. Qazi, O. Khalid, R. N. Rais, I. A. Khan, and A. ur R. Khan, "Optimal Content Caching in Content-Centric Networks," *Wirel. Commun. Mob. Comput.*, vol. 2019, pp. 1–16, 2019.

[13] S. Arif, S. Hassan, and I. Abdullahi, "Cache Replacement Positions in Information-Centric Network," *in The 4th International Conference on Internet Applications, Protocols and Services (NETAPPS2015)*, 2015, pp. 54–58.

[14] J. Ren et al., "MAGIC: A distributed MAx-Gain In-network Caching strategy in information-centric networks," *in 2014 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*, 2014, pp. 470–475.

[15] C. Zhang, C. Xia, Y. Li, H. Wang, and X. Li, "A hotspot-based probabilistic cache placement policy for ICN in MANETs," *Eurasip J. Wirel. Commun. Netw.*, vol. 2019, no. 1, pp. 1–14, 2019.

[16] M. Alkhazaleh, S. A. Aljunid, and N. Sabri, "A review of caching strategies and its categorizations in the information-centric network," *J. Theor. Appl. Inf. Technol.*, vol. 97, no. 19, pp. 4996–5011, 2019.

[17] A. Ioannou and S. Weber, "A Survey of Caching Policies and Forwarding Mechanisms in Information-Centric Networking," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 4, pp. 2847–2886, 2016.

[18] 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.

[19] M. A. Naeem, S. A. Nor, S. Hassan, and B. Kim, "Compound Popular Content Caching Strategy in Named Data Networking," *Electronics*, vol. 8, no. 7, pp. 1–22, 2019.

[20] Q. N. Nguyen 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. 1–18, 2019.

[21] X. Zheng, G. Wang, and Q. Zhao, "A cache placement strategy with energy consumption optimization in information-centric networking," *Futur. Internet*, vol. 11, no. 3, pp. 1–16, 2019.

[22] M. Maizan, F. Hamdi, A. Habbal, N. H. Zakaria, and S. Hassan, "Evaluation of Caching Strategies in Content-Centric Networking for Mobile and Social Networking Environment," *J. Telecommun. Electron. Comput. Eng.*, vol. 10, no. 2, pp. 2–7, 2018.

[23] C. Bernardini, T. Silverston, and O. Festor, "A comparison of caching strategies for content-centric networking," *in 2015 IEEE Global Communications Conference (GLOBECOM)*, 2015, pp. 1–6.

[24] Z. Fan, Q. Wu, M. Zhang, and R. Zheng, "Popularity and gain-based caching scheme for information-centric networks," *Int. J. Adv. Comput. Res.*, vol. 7, no. 30, pp. 71–80, 2017.

[25] M. A. Naeem, R. Ali, B.-S. Kim, S. A. Nor, and S. Hassan, "A Periodic Caching Strategy Solution for the Smart City in Information-Centric Internet of Things," *Sustainability*, vol. 10, no. 7, p. 1–16, 2018.

[26] N. Laoutaris, H. Che, and I. Stavrakakis, "The LCD interconnection of LRU caches and its analysis," *Perform. Eval*., vol. 63, no. 7, pp. 609–634, 2006.
[27] E. J. Rosensweig and J. Kurose, "Breadcrumbs: Efficient, best-effort content location in cache networks," *in IEEE INFOCOM 2009*, 2009, pp. 2631–2635.
[28] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," *in Proceedings of the second edition of the ICN workshop on Information-centric networking*, 2012, pp. 55–60.
[29] K. Cho, M. Lee, K. Park, T. T. Kwon, Y. Choi, and S. Pack, "WAVE: Popularity-based and collaborative in-network caching for content-oriented networks," *in 2012 Proceedings IEEE INFOCOM Workshops*, 2012, pp. 316–321.
[30] Y. Li, T. Lin, H. Tang, and P. Sun, "A chunk caching location and searching scheme in Content-Centric Networking," *in 2012 IEEE International Conference on Communications (ICC)*, 2012, pp. 2655–2659.
[31] C. Bernardini, T. Silverston, and O. Festor, "MPC : Popularity-based Caching Strategy for Content-Centric Networks," *in 2013 IEEE international conference on communications (ICC)*, 2013, pp. 3619–3623.
[32] F. Zhang, Y. Zhang, and D. Raychaudhuri, "Edge Caching and Nearest Replica Routing in Information-Centric Networking," *in 2016 IEEE 37th Sarnoff Symposium*, 2016, pp. 181–186.
[33] J. M. Wang, J. Zhang, and B. Bensaou, "Intra-AS cooperative caching for content-centric networks," *in Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, 2013, pp. 61–66.
[34] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache 'less for more' in information-centric networks (extended version)," *Comput. Commun.*, vol. 36, no. 7, pp. 758–770, 2013.
[35] K. Thar, S. Ullah, and C. S. Hong, "Two layers cooperative caching in Content-Centric Networks," *in Korean Information Science Society*, Korean Institute of Information Scientists and Engineers 2014 Conference, 2014, pp. 1142–1144.
[36] S. Wang, J. Bi, J. Wu, and A. V. Vasilakos, "CPHR: In-network caching for Information-Centric Networking with Partitioning and Hash-Routing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2742–2755, 2015.
[37] Z. Li and G. Simon, "Time-Shifted TV in Content-Centric Networks : the Case for Cooperative In-Network Caching," *in IEEE International Conference on Communications (ICC)*, 2011, pp. 1–6.
[38] A. S. Gill, L. D'Acunto, K. Trichias, and R. Van Brandenburg, "BidCache: Auction-based in-network caching in ICN," *in 2016 IEEE Globecom Workshops (GC Wkshps)*, 2016, pp. 1–6.
[39] J. Kong, L. Rui, H. Huang, and X. Wang, "Link congestion and lifetime based in-network caching scheme in Information-Centric Networking," *in 2017 International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2017, pp. 73–77.
[40] W. Sirichotedumrong, W. Kumwilaisak, S. Tarnoi, and N. Thatphitthukkul, "Adaptive prioritized probabilistic caching algorithm for content-centric networks," *Eng. J.*, vol. 21, no. 6, pp. 11–22, 2017.
[41] X. Hu et al., "An on-demand off-path cache exploration based multipath forwarding strategy," *Comput. Networks*, vol. 166, pp. 1–15, 2020.
[42] D. P. Arjunwadkar, "Introduction of NDN with Comparison to Current Internet Architecture based on TCP / IP," *Int. J. Comput. Appl.*, vol. 105, no. 5, pp. 31–35, 2014.
[43] Zhu, Y., Kang, H., Shi, J., & Yang, J., "A Probability Verification Strategy Based on Credibility of Router in Content-Centric Networking," *Engineering Letters*, vol. 26, no. 3, pp. 396-404, 2018.
[44] "GitHub - mesarpe/socialccnsim," 2019. [Online]. Available: https://github.com/mesarpe/socialccnsim. [Accessed: 04-Nov-2019].
[45] S. Tarnoi, K. Suksomboon, W. Kumwilaisak, and Y. Ji, "Performance of probabilistic caching and cache replacement policies for Content-Centric Networks," *in 39th Annual IEEE Conference on Local Computer Networks*, 2014, pp. 99–106.
[46] C. Bernardini, T. Silverston, and O. Festor, "SONETOR: A social network traffic generator," *in 2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 3734–3739.
[47] B. Li, L. Rui, X. Qiu, and H. Huang, "Content caching strategy for edge and cloud cooperation computing," *in 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 2019, pp. 260–265.
[48] C. Shan and J. Luo, "Node importance to community-based caching strategy for information-centric networking," *Concurrency and Computation: Practice and Experience*., vol.31, no. 21, pp. 1–11, 2019.
[49] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 3, pp. 1473–1499, 2015.