IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 13, 2018, accepted September 23, 2018, date of publication October 1, 2018, date of current version October 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2872614

Performances of Probabilistic Caching Strategies in Content Centric Networking

MUHAMMAD ALI NAEEM[1], **SHAHRUDIN AWANG NOR**[1],
SUHAIDI HASSAN[1], **(Senior Member, IEEE), AND**
BYUNG-SEO KIM[2], **(Senior Member, IEEE)**
[1]InterNetworks Research Laboratory, School of Computing, University Utara Malaysia, Sintok 06010, Malaysia
[2]Department of Software and Communications Engineering, Hongik University, Sejong 30016, South Korea

Corresponding author: Byung-Seo Kim (jsnbs@hongik.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) through the Korea Government under Grant 2018R1A2B6002399.

ABSTRACT Information centric network (ICN) is progressively becoming the revolutionary paradigm to the traditional Internet with improving data (content) distribution on the Internet along with global unique names. Some ICN-based architecture, such as named data network (NDN) and content centric network (CCN) has recently been developed to deal with prominent advantages to implement the basic idea of ICN. To improve the Internet services, its architecture design is shifting from host-centric (end-to-end) communication to receive-driven content retrieval. A prominent advantage of this novel architecture is that networks are equipped with transparent in-network caching to accelerate the content dissemination and improve the utilization of network resources. The gigantic increase of global network traffic poses new challenges to CCN caching technologies. It requires extensive flexibility for consumers to get information. One of the most imperative commonalities of CCN design is ubiquitous caching. It is broadly accepted that the in-network caching would improve the performance. ICN cache receives on several new characteristics: cache is ubiquitous, cache is transparent to application, and content to be cached is more significant. This paper presents a complete survey of state-of-art CCN-based probabilistic caching schemes aiming to address the caching issues, with certain focus on minimizing cache redundancy and improving the accessibility of cached content.

INDEX TERMS Content centric networking, leave copy everywhere (LCE), caching, probabilistic caching, content delivery networks (CDN).

I. INTRODUCTION

Internet usage has increased extensively in the past decade, especially regarding the broad use of Video on Demand (VoD), which increases Internet traffic. The current Internet architecture supports end-to-end communication, and the content retrieval process occurs in two steps. First, the name resolution provides identifiers for the content (i.e., URL) with the location (i.e., IP address), and second, consumer Interests are transmitted based on location from consumer to provider and Interested content from provider to consumer [1]. The name resolution occurs for the content at the application layer before forwarding the content which wastes network resources. For instance, if a copy of the interested content resides at network router near the consumer, the consumer's Interest does not need to be transmit to the main content provider, which decreases the usage of network resources and network traffic [2]. Moreover, the Internet is progressively recycled for information distribution with, relatively less use for end-to-end communication between end hosts [3]. Although Internet requirements have been increasing the evolution of Internet architecture has not followed the growing demands of Internet consumers and communications data [4]. For example, the existing communication worldview of the Internet presents numerous restrictions. First, the current Internet design offers numerous duplicates of objects, yet these duplicates are not connected. Keeping in mind the end goal to share the system's assets, extra overlay instruments have been proposed over the TCP/IP stack [5]. Reliable and fast object exchange requires application specific systems, for example, Content Delivery Networks (CDNs) or Peer to Peer (P2P) [6] administrations. Second, security is accomplished via outsider applications and administrations. Establishing trust in receiving objects is difficult and the majority of associations depend on suspicious locations. Third, actually receiving objects is strongly related to the object's location. Each Internet packet follows source and destination addresses. Every time an Interest packet is sent for object accessing, it needs addresses [7]. These Interests are firmly connected to an address, flagging a specific object location, even though consumers do not care about the location and just want the object to arrive as fast as possible. The direct manner to resolve this problem is to replace the "where" with the "what" [8]. Host-to-host correspondence was used to take care of the issues in the 1960s. Researchers contend that architecture using named objects is a better fit for the communication needs of today.

According to the Visual Networking Index (VNI) report, researchers have recognized other issues related to the IP network architecture that require quick comprehension [9] to have an acceptable platform that includes data integrity, confidentiality, availability, accountability (Owner/Publisher identification and authentication), and especially data transmission [10]. Several projects and proposals have been developed to improve Internet architecture, including Information-Centric Networking (ICN), which is a leading research focus for future Internet paradigm [8]. A number of ICN projects have been formed, such as the European Union Research and Innovation program (EU FP7). Under FP7 [11], several projects are being developed, for instance, Network of Information (NetInf) [12], 4WARD [13], Publish/Subscribe Internet Routing Paradigm (PSRIP) [10], [14], Publish/Subscribe Internet Technology (PURSUIT), Comet [9], COMBO, SAIL project, CONVERGENCE [14], [15], COAST [16], and Green ICN [17].

Similarly, the US research community has also initiated a number of ICN-based projects, including Content-Centric Networking (CCN), Named Data Networking (NDN), and Data-Oriented Network architecture (DONA). Another ICN-based project known as Active Content Management at Internet Scale (COMIT) was launched at University College London (UCL) in January of, 2014 [18]. All of these projects feature different functionalities and implementations. However, the basic purpose of all ICN-based projects is to design a network architecture that supports the named data distribution to handle the flash-crowd effect, disruption, and denial of service; reduce the use of resources and decrease energy consumption [19]. In addition, CCN provides the complete functionalities of ICN.

To summarize, all of the ICN projects have been developed by combining several modules in which caching is considered the most dominant component for actualizing the ICN manageability [20]. The Cache is used to store the transmitted content in two different forms: on-path and off-path [21]. These caching techniques can be implemented efficiently with the help of content deployment strategies. In addition, a basic need behind caching performance is determining the appropriate position to place the transmitted content. Consequently, many cache deployment strategies have been proposed for achieving some advantages, such as proper use of bandwidth by enhancing the appropriate delivery of information and reduction of delays [22], as well as minimization of the overall load on the main server (content provider) [23].

Several surveys recently published related to CCN caching strategies, provide information regarding the caching decisions for disseminated data content [24]–[26]. However, the present survey is conducted to determine caching performance based on the most relevant basic metrics for improving the same. Moreover, this paper focuses on the critical analysis of probabilistic caching strategies and the impact of basic caching evaluation metrics, including content redundancy, content diversity, Stretch, and, most importantly, cache hit ratio. In addition, authors conducted simulations to compare a number of probabilistic strategies within the same environment to find which one is the most beneficial to meeting high performance with all of the mentioned key metrics. Fig. 1, demonstrate the classification of all caching strategies mentioned in this article. Section II gives an overview of CCN architecture. Section III presents the basic caching performance metrics to which this survey is directed. In section IV, we discuss the probabilistic caching strategies and their basic architectures. In section V, we present the simulation results of the mentioned strategies. Section VI, provides a critical summery of the compared strategies and, finally, we conclude the paper in section VII.

II. CONTENT-CENTRIC NETWORKING

According to the latest forecast [27], global IP traffic will attain more than 1,000 exabytes (EB) per year in 2018. The majority of this increased traffic comes from peer-to-peer (P2P) IP infrastructures and a variety of different forms of video traffic, such as TV and VoD, which accounts for almost 90% of global customer IP traffic. The global mobile traffic is also estimated to increase at extreme rates at the same time. To deal with this enlargement in terms of data quantity and devices, one solution is to organize application layer overlays, such as CDNs [28] and P2P applications, that cache content, offer location-free access to data, and enhance the data delivery method. This payoff should make access to named data or objects available, with replicated web resources, rather than the conventional host-to-host data delivery model. However, these procedures would reside in networks, and understanding the full potential of content-based sharing in today's IP-based platform is very complicated. CCN is an innovative networking approach to facilitate named data or objects as the supreme network entities [29]. It supports object naming and pervasive in-network caching to supply well-organized and vigorous network services [30]. CCN will modify addresses from 1 billion IPs to at least 1 trillion content names [31].

CCN can be considered an enhanced generalization of CDN technology, and CCN operation is sufficient on a scale similar to CDN [32]. Moreover, CCN [33] is not limited to only media-sharing scenarios; it includes other schemes as well, such as data collection. For data dissemination,

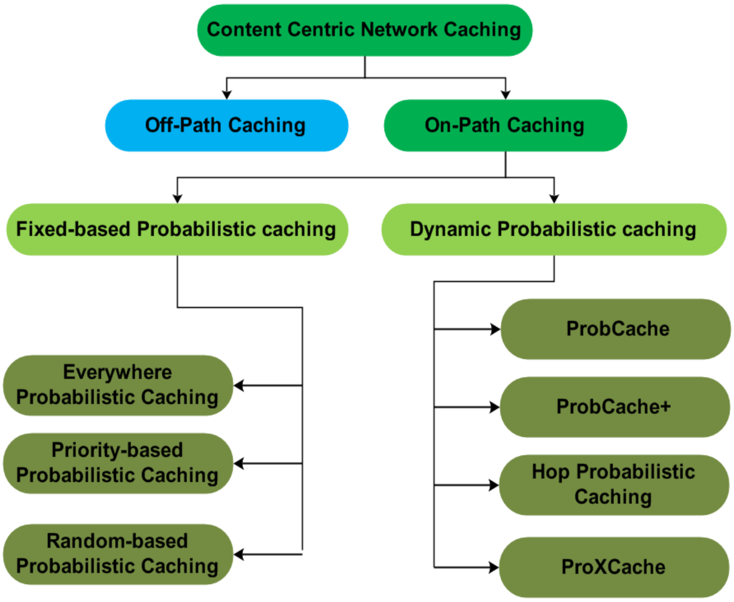


FIGURE 1. Caching classification in CCN.

two types of message packets are used in CCN: Interest and Data (content). The Interest message with the name of desired content transits the consumer's interest [34]. The Data, which retrieved from content provider, is known as content in CCN. An Interest message identifies the Data packet (content) and retrieves it by specifying a full name with other restrictions that designate acceptable data. A Data packet (content) message holds the required information payload and the identification of the provider. Fig 2, illustrates a CCN node consisting of three types of data structures such as; Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB) which is used to facilitate ubiquitous caching and loop-free forwarding. CS indicates the cache storage that holds the content for subsequent retrieval. It is managed via cache placement/replacement algorithms [35] that exploit the possibility of data reuse. The PIT is used to record unsatisfied (pending) interests. Each entry in the PIT has a list of source entries incorporated with incoming faces to the CCN node. The PIT has a specific timeout interval for unsuccessful Interests to avoid maintaining previous Interest records. The FIB is used to sustain the topology structure in tabular form to facilitate outbound face information for Interests. These tables are used for longest-match lookups in the prefix order through the content name. Each FIB entry presents a set of corresponding faces rather than a single face. ICN construction was developed in Europe. PSIRP/PURSUIT [36] sends forwarding information in packets, whereas CCN caches routing information in CCN nodes. In CCN, Interest packets are forwarded all the way through FIB, but the Data packets are forwarded according to the PIT records, which direct Data packets back to consumers. Consequently, an outstanding FIB design is necessary for transmission of both Interest and Data packets in CCN [37]. In Fig 2, the CCN content caching mechanism is demonstrated as Consumer A sends out Interest1 for Content C1. C1 is found at router R1 because the content provider already published C1 at R1; therefore, R1 now becomes the intermediate provider. Consequently, R1 instantly responds to Interest1 by sending the desired Content C1, and a copy of C1 is cached locally during its transmission at all on-path routers (R2, R3, and R4). A subsequent Interest2 is generated from Consumer B to retrieve C1. As a result, C1 is found at R2 at this time, and R2 sends C1 to Consumer B with multiple caching operations along the data routing path (R5 and R6), as illustrated in Fig 2.

A. CACHING STRATEGIES IN CCN

An unbelievable expansion of Internet traffic and frequent transmission of identical content have resulted in problem with server load, bandwidth use resource usage, and response times. These issues will only increase and will be difficult to solve in the future through the current IP-based Internet architecture [38], [39]. Internet usage is growing daily, and adequate resources are not available to meet the huge and constantly growing needs [39]. To enhance Internet performance, researchers want to implement the best possible approach that will overcome these rising difficulties. Moreover, the Internet needs modification to its architectural design to meet predicted future usage requirements [40]. In this case, CCN offers a better way to resolve the aforementioned issues by implementation of in-network caching, which is fundamental to and unavoidable in all CCN architectures [41].

In-network caching is used to diminish the average cost (energy and time) and enhance the content retrieval procedure, because a cache stores a copy of frequently interested data temporarily for subsequent Interests. A prominent example of this storage is web caching, in which web documents

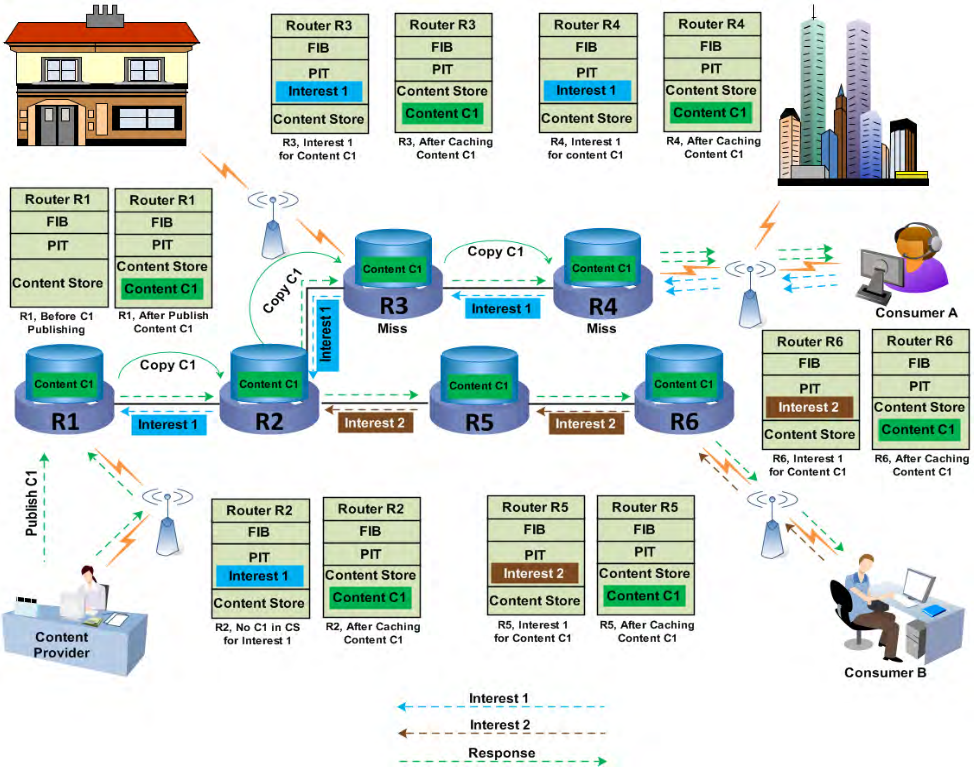


FIGURE 2. Content-centric networking.

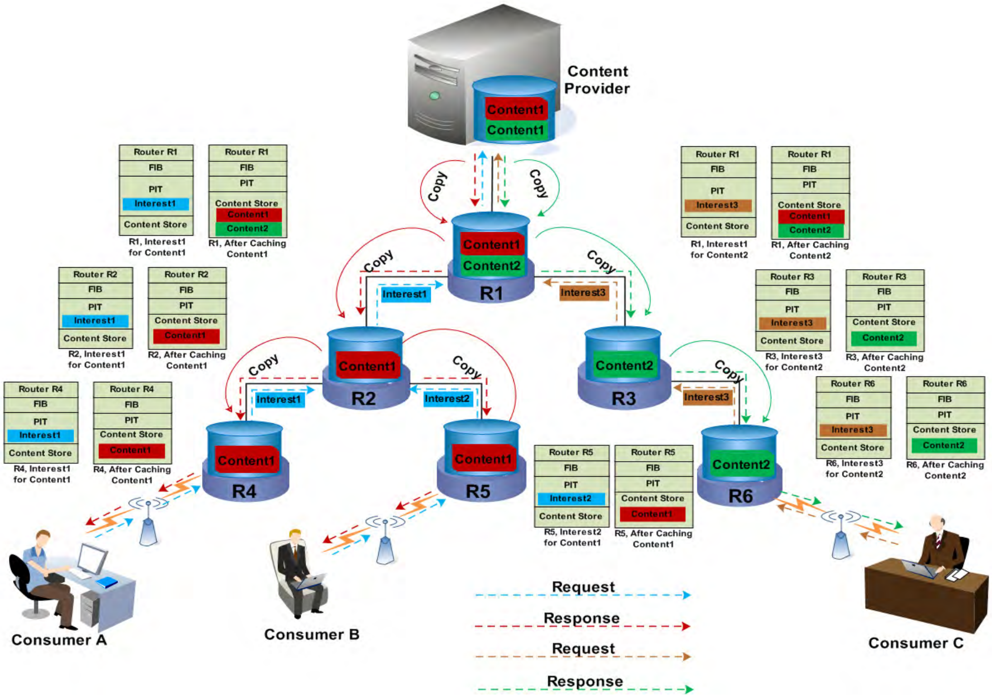


FIGURE 3. Caching mechanism in CCN.

(e.g., HTML pages and images) are stored to mitigate issues in bandwidth use, server load, and reply time. Caching makes CCN different from the currently IP-based Internet, where the cache is used to transmit content to the consumers next CCN cache minimizes the response time, lessens resource consumption, and boosts data communication services [42], [43]. In Fig. 3, Consumer A sends Interest1 for a specific data Content1 to its nearest router (R4) with the corresponding content in its cache [44]. The intermediate routers become content providers whenever disseminated content

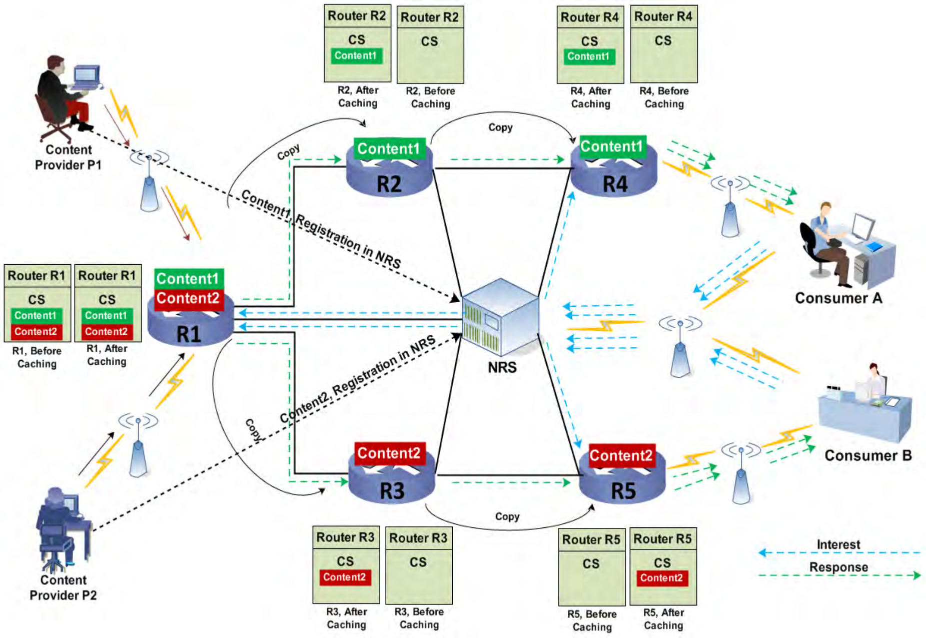


FIGURE 4. Off-path caching.

passes through their cache, and they store the passing content locally in their cache for subsequent Interests. Therefore, R4 sends Content1 to its desired Consumer A. The caching strategies are divided into various categories based on their characteristics. It is hard to find an appropriate criterion that will best meet all future requirements. As stated previously, CCN caching is divided into Off-path caching and on-path caching, as described next.

1) OFF-PATH CACHING

Off-path caching is similar to the traditional proxy caching or CDN server. The retrieval of the content in off-path caching requires redirection of Interests. For off-path [43] caching a special entity, called the name resolution system (NRS), must be integrated. All sets of content are registered in NRS at the time of creation, and consumers can send their Interests for registered content [45]. The general problem with off-path caching is choosing what to cache and where to cache it. The other issue is how to reduce the overhead to inform the NRS when a new item is cached or an old item is discarded. The exact information about the content depends on the NRS, which needs to be updating continuously with all the information related to the registered content. In off-path caching, a consumer sends an Interest to the NRS, and the NRS compares it with the available content information [44]. Afterwards, the Interest is forwarded to the corresponding provider (router), and the provider sends the required content to the consumer following the returning path, as shown in Fig. 4. As the content is registered in NRS, the consumers send their Interests to the NRS. After that, the NRS forwards the received Interests to the appropriate providers (e.g., router or main content provider P1 and P2). The subsequent content is sent to the desired consumer through NRS or directly to the consumers. In Fig. 4, Consumers A and B send their Interest packets to the NRS to retrieve Content1 and Content2. The NRS redirects the received Interests to the appropriate content provider (e.g., router R1). R1 responds by sending the interested Content1 and Content2 to the appropriate Consumers A and B. The transmitted sets of content are cached along the data routing path during their dissemination from providers to the consumers; for example, Content1 is cached at routers R2 and R4, and Content2 is cached at R3 and R5, respectively, as illustrated in Fig. 4.

2) ON-PATH CACHING

In the on-path approach, when a caching node entertains a consumer's Interest, it responds to the interested consumer with a locally cached copy without NRS participation [46]. Therefore, this approach diminishes the computation and communication overhead during insertion of transmitted data in the network [9], [47]. When the Interest is projected in network, the dominant provider (router) instantly responds to the received Interest by sending a copy of locally cached content to the appropriate consumers. Fig. 5 illustrates the caching scenarios as follows: Consumer A sends out an Interest to retrieve Content1, which is found at router R1. Consequently, provider router R1 instantly responds to Consumer A by sending interested Content1, and a copy of the demanded content is cached at all intermediate routers (R2 and R4) during its transmission from provider router R1 to Consumer A.

III. PERFORMANCE METRICS

How each node makes a caching decision is a major problem in on-path caching that impacts the performance of the

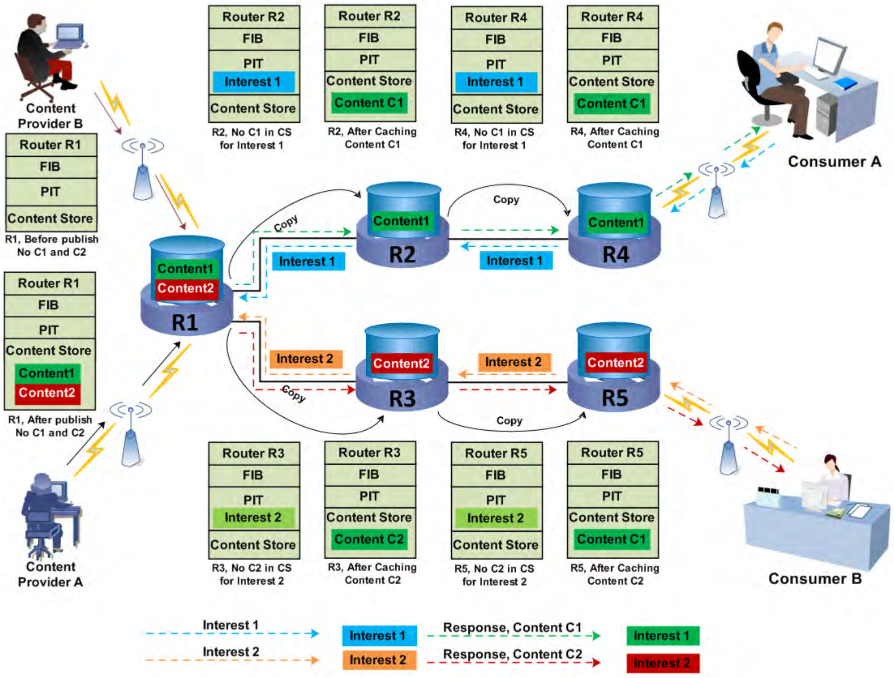


FIGURE 5. On-path caching.

content delivery process. For example, popular content needs to be cached at the node where it will be demanded next [48]. The information content can be cached opportunistically at the delivery path for the consequent Interests [49]. The efficiency of caching is dependent on different types of performance metrics that are necessary to improve the content caching strategy by considering the cache size, which is small compared to the communicating data content. Therefore, some essential performance metrics are included in this study to critically compare types of probabilistic caching strategies and determined which is the optimal caching.

A. CACHE HIT RATIO

The cache hit ratio is the portion of content Interests satisfied by caches that are implemented within the network. Several studies in the database and networking domain have emphasized the performance of the network by improving the cache-hit ratio. This metric shows the potential of the caching strategy to moderate the amount of redundant content.

B. CONTENT REDUNDANCY

Content redundancy is a significant metric to assess homogeneous data dissemination in CCN. In CCN, redundancy is the number of content copies cached at more than one location in the network with fewer referrals or less unsubscribing [50]. Redundancy is therefore presented for each network setup and accordingly during Interest and content dissemination.

C. CONTENT DIVERSITY

This defines the number and rate of caching distinct content in the heterogeneous network or the ratio of how different sets of content accumulate in the network cache. The resulting rate is computed to mitigate high information replication [51]. Diversity defines the performance of cache replication. It provides and computes the ratio of distinctly stored unique sets of content in the cache-able devices.

D. STRETCH

Stretch refers to the number of hops (routers) that need to be covered by a consumer's Interest between the content provider (where the hit occurs) and the consumer. Whenever the consumer sends an Interest for some content, the Interest needs to be covered by a number of hops to reach the source for the required content. While the Interest can find a copy of the required content from any of the routers that appears on the path, sometimes it has to go to the main content provider server, which increases the average cost. Therefore, each caching strategy tries to reduce the maximum number of hops between providers and consumers.

IV. CCN CONTENT CACHING STRATEGIES

Cache deployment deposits the consumer's desired content along the data routing path in provisional storage (cache) that is allocated within the network nodes. Caching plays an immense role in network performance by diminishing bandwidth consumption and reducing latency when accessing the required content [52]. Caching also minimizes the server load while maximizing the availability of the content. An essential operation of these caching strategies is forwarding the desired content to the interested consumers and caching a copy of the content near the interested consumer to minimize the average cost for subsequent Interests (requests). Recently, a number of

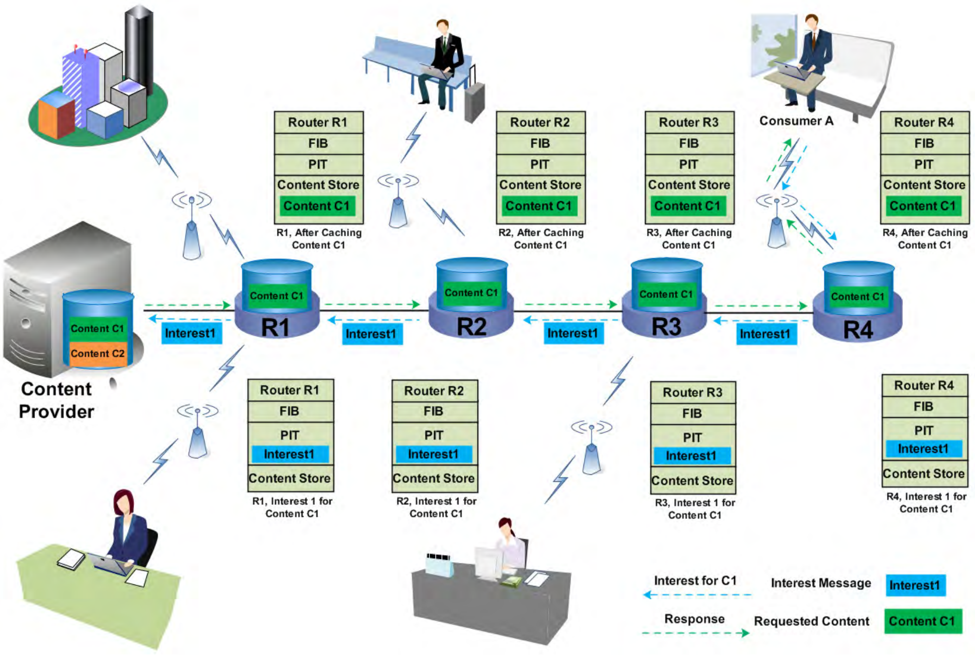


FIGURE 6. Everywhere probabilistic caching.

content deployment strategies have been projected to deploy CCN in-network caching. These strategies boost information replication and decrease reaction time [53], [54]. In addition, the strategies use fewer network resources and reduce network traffic [55].

CCN's basic problem is that the capacity of in-network caches is much smaller than what is needed to accommodate all arriving content. Additionally, proficient content dissemination can be achieved by integrating a huge capacity of caches, but that is not a straightforward way to supply a large quantity of caches. A system for efficient content dissemination needs to be designed that will broadcast the content in a disciplined manner with less use of resources (caches). In the conventional CCN caching strategies, when a consumer shows Interest for particular content, then a copy of the desired content is cached at all the routers that are available along the publish-subscribe path. As a result, the subsequent Interests may be fulfilled locally [56]. Various kinds of caching strategies all with differing implementation details have been constructed to comprehend the main target of CCN caching. The content caching strategies depend on different criteria, such as caching decisions based on distance from the consumer, centrality-based caching, probabilistic caching based on the popularity of frequently interested content, label-based caching, and caching based on a mathematical equation. Moreover, the caching according to naming granularity can be categorized into the following types [57]: chunk level caching, object level caching, and packet level caching. Of all the strategies mentioned, researchers are most interested in probabilistic caching, which has a number of advantages due to its vast domain [3]. For example, probabilistic caching [58] is a well-known content caching strategy, and the research community captures more interest because of its adjustable probabilistic value [59], which is allocated according to the manufacturer. Probabilistic caching is separated into two sub-categories. The primary subcategory is fixed probabilistic caching (FIX), and the secondary subcategory is dynamic probabilistic caching.

A. FIXED-PROBABILISTIC CACHING (FIX)

This sort of content-deployment mechanism is based on the pre-determined probabilistic values to cache the consumer's interested content along the data delivery path [60], [61]. Caching of the content is taken individually at all routers without the involvement of collaboration among the nodes. FIX has no distinction regarding the transmitted content [62].

1) EVERYWHERE PROBABILISTIC CACHING (EPC)

The Everywhere Probabilistic Caching (EPC) strategy lies in FIX and is based on probabilistic situations. According to probability, the demanded content is cached with probability at all the nodes that have empty caches along the downloading path. If the probability value is set to 1, then it works just like CCN default caching, that is, Leave Copy Everywhere (LCE) [63]. If it has a value other than 1, the disseminated content will be stored according to the probability value (e.g., probability based on strategy and the available cache). Fig. 6 shows the dissemination of two sets of content through different probabilities [57]. If the caching is set to probability 1 at that moment, the content will be cached at all routers to facilitate the available cache on the delivery path (see Fig. 6). In Fig. 6, Consumer A sends out an Interest1 to the network to retrieve content C1, but the desired content is not found in any router's cache so Interest1 reaches the content provider.

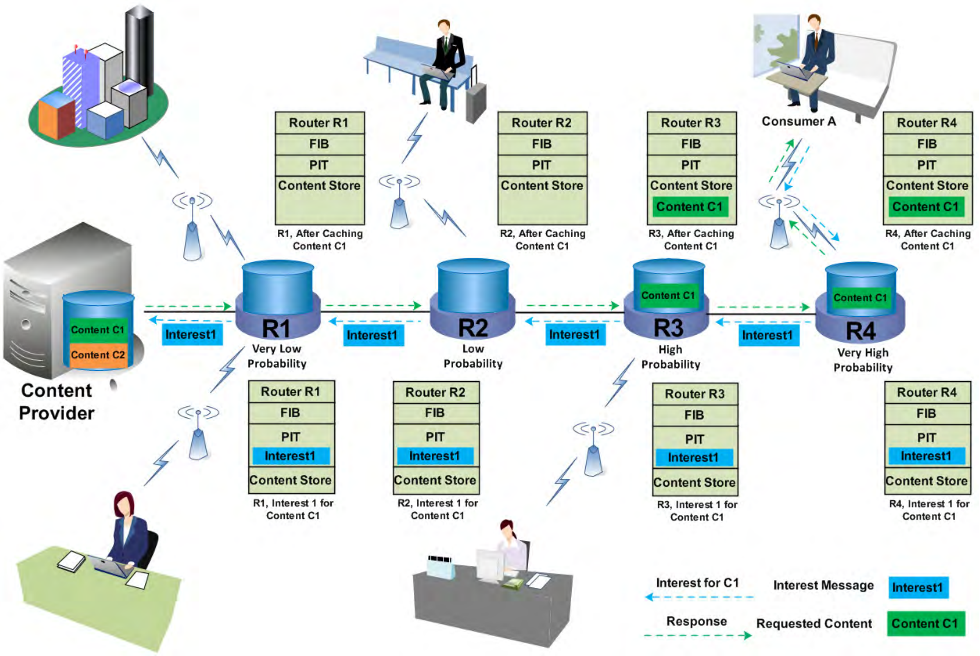


FIGURE 7. Priority-based probabilistic caching.

Consequently, the provider responds to received Interest1 by sending the required content C1 to Consumer A, and a copy of C1 will be cached at all on path routers (R1, R2, R3 and R4). Moreover, for the subsequent Interests, the C1 will be send from the nearest router R4.

2) PRIORITY-BASED PROBABILISTIC CACHING (PBPC)

The Priority-Based Probabilistic Caching (PBPC) strategy lies in FIX and assigns the priorities to the content based on the stretch between the content providers and the consumers. Probabilities for the content vary at all nodes, and the disseminated sets of content are cached along the consumer's path based on their probabilistic value [3], [57]. The probabilistic value of the content is inversely proportional to the distance. Therefore, if the distance between content provider and consumer is farther, the content will be indicated as a low probability to be cached along the data delivery path. On the other hand, if the distance between the provider and consumer is shorter the interested content will be posed as a high probability to cache the disseminated content along the data delivery path [63].

Fig. 7, provides a visual of the entire process of PBPC, in which Consumer A sends an Interest1 to get Content C1, and C1 is found in the content provider's cache. As a result, the desired content is sent back to Consumer A, and a copy of C1 is locally cached along the delivery path. According to PCPB, the probability is divided to cache the disseminated content based on distance. If the distance is small, the router gets more chances to cache the content; however, as the distance increases, the probability decreases. In Fig. 7, interested content C1 is cached only at routers R3 and R4 because R3 and R4 show the highest probability to cache C1 due to the shorter distance from Consumer A. On the other hand, routers R1 and R2 are located far from Consumer A, and both have low probability to cache C1 within their cache storage. Consequently, the content does not have much probability to cache at R1 and R2. Therefore, C1 only cached at R3 and R4 because of the short distance between Consumer A and these routers (R3 and R4) [63], [64].

3) RANDOM-BASED PROBABILISTIC CACHING (RPC)

Random-Based Probabilistic Caching (RPC) is derived from FIX, in which the caching decision of disseminated content is determined based on the randomly chosen probabilistic value [65]. RPC offers two advantages: (1) exhibits simplicity in structure, and (2) extracts low overhead when searching content for caching. On the other hand, RPC possesses low efficiency due to its uncertain and random nature. This mechanism is used as a benchmark to compare the performance of latest establishing caching strategies [66]. In this strategy, a copy of the required content is cached at randomly selected locations along the data delivery path from provider to consumer.

RPC can be materialized into several kinds of approaches; for example, the caching decision can be performed based on the levels of tree structure topology. The caching depends on the algorithm regarding which levels of the caching router are available [67]. For instance, if level-1 is chosen for caching the transmitted content, then it will cache only at routers located in level-1 of the tree topology [68]. On the other hand, if level-2 is selected for content caching, the content will be cached at the routers located at level-2 of the tree topology

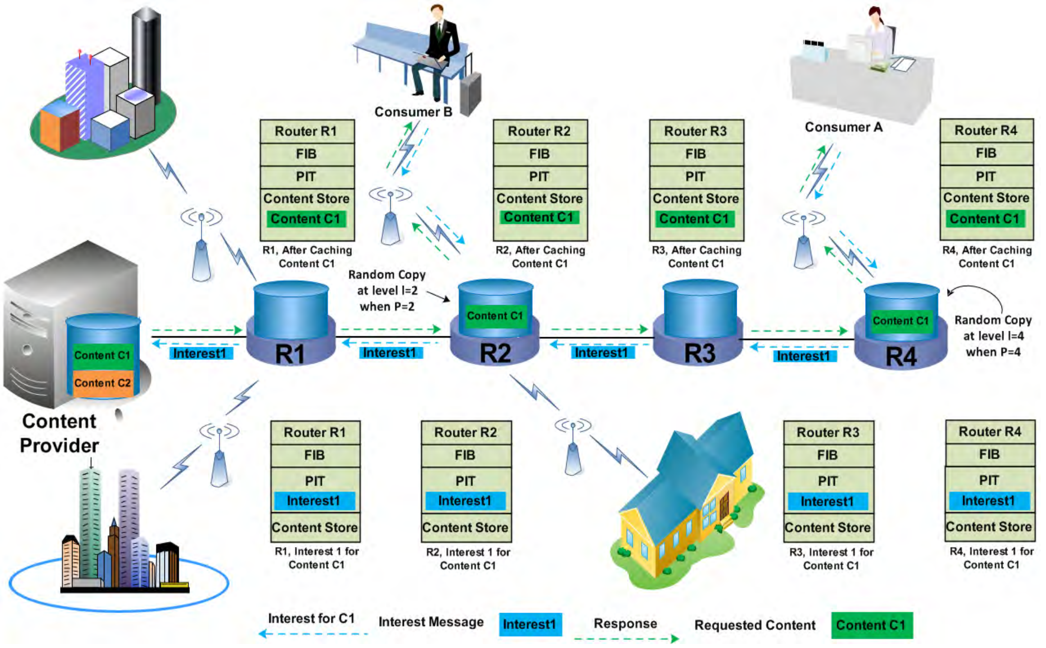


FIGURE 8. Random-based probabilistic caching.

(see Fig. 8), and when Consumer A sends out an Interest to retrieve content C1, C1 is cached at router R2. If level 4 is chosen, for example, the content will be cached at R4, as illustrated in Fig. 8. The caching decision is predefined in the strategy algorithm, and it can be changed according to the nature of the environment.

Moreover, it can be changed on the basis of data traffic congestion and to reduce the distance between providers and consumers for subsequent Interests. RPC provides a simple structure and low overhead, but it becomes questionable due to the unpredictable nature of randomness and its different types of random structures. RPC does not provide distinct position, which causes high retrieval times because there is a possibility of caching content far from the consumer, increasing the distance between providers and consumers.

B. DYNAMIC PROBABILISTIC CACHING

The focus of dynamic-probabilistic caching strategies is to enhance the structure of network topology with traffic patterns. The dynamic caching strategies are developed to diminish network and caching redundancy to mitigate the resource usage rates on the whole network. Caching decisions in dynamic probabilistic caching is dependent on the capacity of the routers' caches, the amount of data traffic, and the time required for content to be cached [69]. All content are incorporated with a time window to facilitate how much time they can be cached at a specific position; this interval is calculated via the Target Time window (Ttw) [70]. Moreover, two primitives, Time Since Inception (TSI) and Time Since Birth (TSB), are used for the calculation of path length. TSI is used within the Interest header, and TSB is assigned within the content header. When a consumer sends an Interest packet for some content, the value of TSI is initiated to 1, and the value increases by one when it passes through one hop. TSB is coupled with desired content. When interested content is hit at a provider, the value is initialized with 1, and this value is boosted up as the content traverses to the consumer. When a hit occurs, the TSI header of the Interest merges with content. However, its value remains unaffected as the content passes through the data routing path to the consumer [71]. TSI indicates the path length of the Interest from consumer to provider, and TSB represents the distance from provider to consumer [3].

1) PROBCACHE

ProbCache is a type of dynamic probabilistic caching [31] in which the caching decision is based on the cache capacity along the data downloading path and the factor used to calculate the tendency to cache the content close to the consumer [72]. ProbCache aims to improve the caching tendency of the consumer's interested content to have it cached nearer to consumers. The probabilistic value for content caching along the data routing path can be calculated as

$$\text{ProbCache} = \text{TimesIn} + \text{CacheWeight} \tag{1}$$

$$\text{TimesIn} = \frac{\sum_{n=1}^{x-y+1} Nn \ldots}{\text{Ttw}{*}\text{Nx}} \tag{2}$$

$$\text{CacheWeight} = \frac{\text{y}}{\text{x}} \tag{3}$$

where N_n indicates the cache storage of all the nodes along the path between source and consumer, N_x is the average cache capability of all the routers along the delivery path, Ttw is the Target Time window assigned by the mechanism

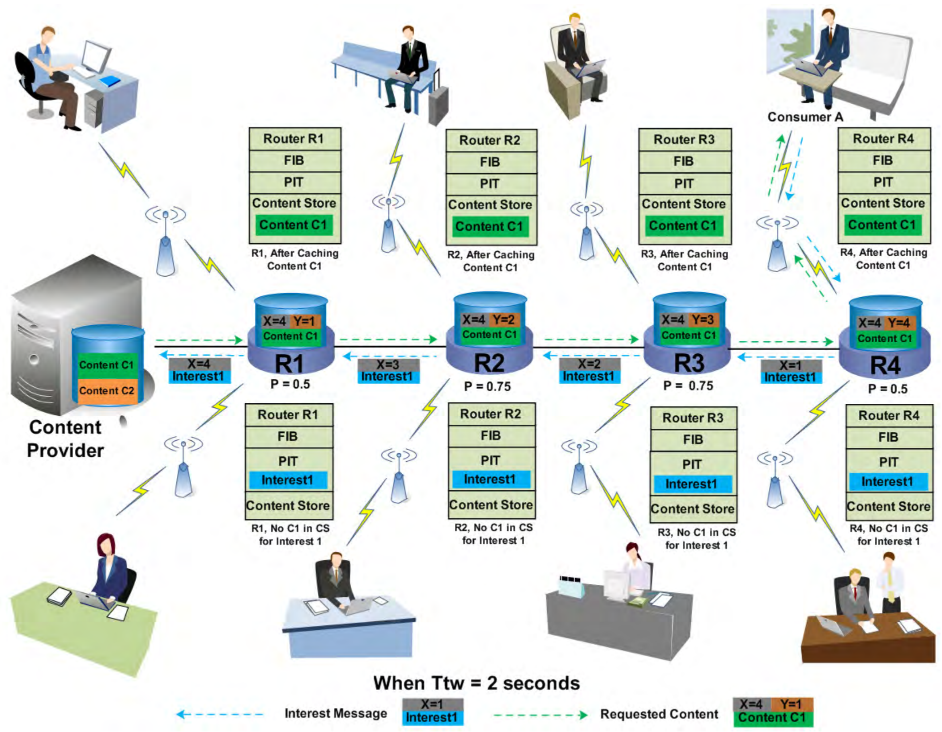


FIGURE 9. ProbCache.

that gives details about how much time content can be cached along the data routing path, and n shows the quantity of routers along the path. Moreover, x and y stand for the Interest header with TSI and interested content header with TSB, respectively. In Eq. 2, y symbolizes the TSB header in which the number of nodes is calculated between provider and consumer, whereas x illustrates the TSI header coupled with the consumer Interest that quantifies the number of routers between consumer and provider. These equations need to be calculated at all router along the delivery path throughout content transmission [73]. However, the time period used to cache content at the subscriber path can be calculated by the Times-In factor. At the same time, the CacheWeight factor is used to cache content close to the consumer.

This caching strategy yielded poor performance for various caching spaces along the path. Fig. 9, demonstrates the design of ProbCache: When Consumer A sends away an Interest to the content provider, at that moment, the value of the TSB header is incremented by 1 at each router. While the source acts in response to the Interest via sending interested content C1, at that time, the TSI value is attached along with the TSB header of the desired content. Although the TSI remains unchanged throughout content delivery, the value of TSB is raise incrementally at all nodes. If both values will be the same according to ProbCache, the interested content will be cached near Consumer A at the edge router R4.

According to ProbCache, the Ttw is quite small, which increases the content eviction rate and decreases the caching gain. When Consumer A throws an Interest for content C1, the value of x is added to at every router along the routing path from Consumer A to the provider. Subsequently, the provider responds to Consumer A by sending content C1; the value of x is attached within the content header, while the value of y is added to along the data routing path at all the routers (from provider to Consumer A). If the value of Ttw is agreed to as Ttw $= 2$ sec for an identical cache size followed by calculating the TimesIn factor and CacheWeight factor, the probabilistic values would be equivalent to $p = (0.5)$ R1, $p = (0.75)$ R2, $p = (0.75)$ R3, and $p = (0.5)$ R4, alongside the path (where R1, R2, R3, and R4 are the caching routers alongside the subscriber path). Obviously, the probability is higher to cache content at the symmetric routers than at the edge routers [74]. Therefore, ProbCache dispenses a fair cache distribution along the delivery path.

2) PROBCACHE+

ProbCache+ [70] is the enhanced caching mechanism version of ProbCache that was developed to improve the unfair cache allocation among disseminated content. In this strategy, the cache allocation was enhanced by changing the value of Ttw along with the routers at the delivery path. To support this, an improved form of CacheWeight has been projected. According to ProbCache+, the value of Ttw is decreased by increasing the distance from consumers; in other words, Ttw is inversely proportional to distance [75]. The probability is amplified gradually along the delivery path as the content caches near the consumer and presents a high probability of maintaining its caching situation for a long time. The

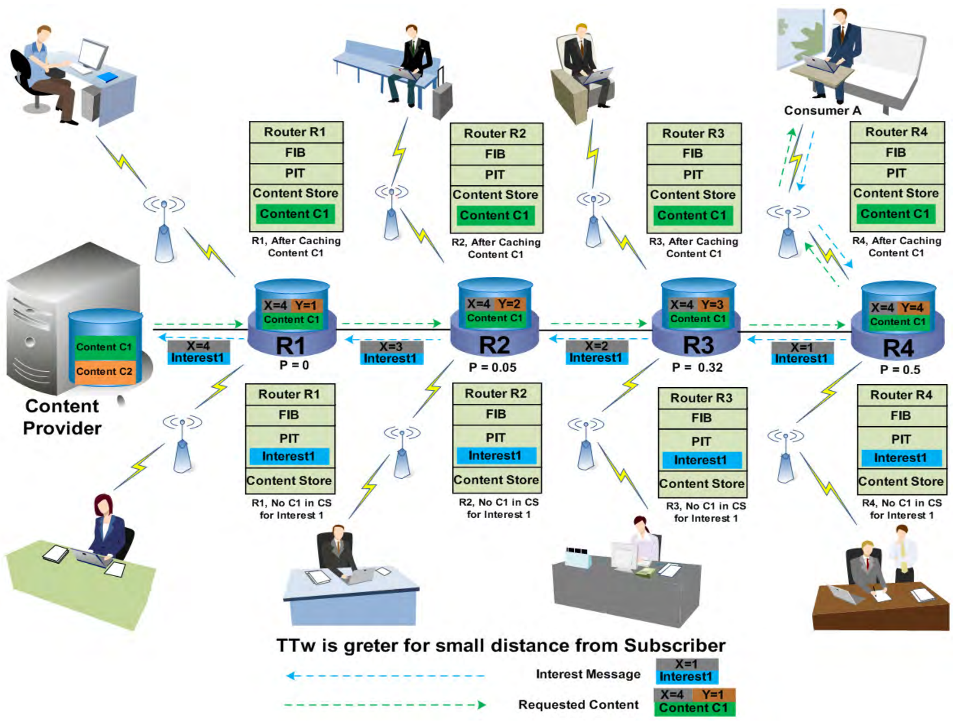


FIGURE 10. ProbCache+.

probabilistic value for content caching can be calculated as

$$\text{ProbCache+} = \text{TimesIn} \times \text{CacheWeight} \tag{4}$$

$$\text{TimesIn} = \frac{\sum_{n=1}^{x-y+1} Nn \ldots}{\text{Ttw} * \text{Nx}} \tag{5}$$

$$\text{CacheWeight} = \left(\frac{\text{y}}{\text{x}}\right)^{x} \tag{6}$$

In these equations, Nn is the cache space of all the routers along the data routing path, and N_x is the average cache ability of the routers. Ttw is the Target Time window set down by the strategy regarding the length of time interval a content can be cached at the consumer's path, n is used to count the number of routers, and x and y symbolize the Interest header TSI and content header TSB, respectively. To improve this caching, the CacheWeight Factor is modified in exponential form to calculate the probability of caching content at all on-path routers, as demonstrated in Fig. 10. ProbCache+ enhances the ProbCache algorithm by distributing the probabilistic value according to the distance in hops from consumers; for example, for Consumer A, it will be (0, 0.32, 0.05, 0.5), along with router R1, R2, R3, and R4, respectively. This proves that the probability is rising toward the consumer and falling away from the provider. If the distance is long, the probability will be reduced. ProbCache and ProbCache+ aim to tackle the problems related to inefficient resource usage in different ways. They share the resources (cache) fairly among the content, but these strategies seem unable to achieve their goals completely.

Both caching strategies introduce a number of parameters that have no content distinction and need to be set arbitrarily. In addition, both strategies need to update the consumer Interests and content in the form of TSI/TSB, which increases the computational cost and memory consumption at all routers. Moreover, these caching strategies introduce the Ttw by using a special variable that also increases the computational cost. Finally, both strategies do not distinguish cached sets of content and provider no criterion to handle the frequently interested content [58].

3) HOP-BASED PROBABILISTIC CACHING (HPC)

Hop-Based Probabilistic Caching (HPC) [76] was developed to conquer the rising issues in ProbCache+. It was created by extended two factors: The primary factor is CacheWeighty, and the succeeding factor is $\text{CacheWeight}_{\text{MRT}}$.

The CacheWeighty factor is used to diminish content redundancy by gradually pushing the content to consumers, and the y parameter is used to calculate the number of hops along the back path source to the consumer. The probabilistic value for content caching along the data routing path can be calculated as

$$\text{HPC} = \text{CacheWeighty} + \text{CacheWeight}_{\text{MRT}} \tag{7}$$

$$\text{CacheWeighty} = \frac{1}{y+\alpha}\alpha \geq 0 \tag{8}$$

$$\text{CacheWeight}_{\text{MRT}} = \text{MRT}_{\text{m}} + \text{MRT}_{\text{exp}} \tag{9}$$

The CacheWeighty factor is equivalent to the distance between the router and the source. The distance is taken in

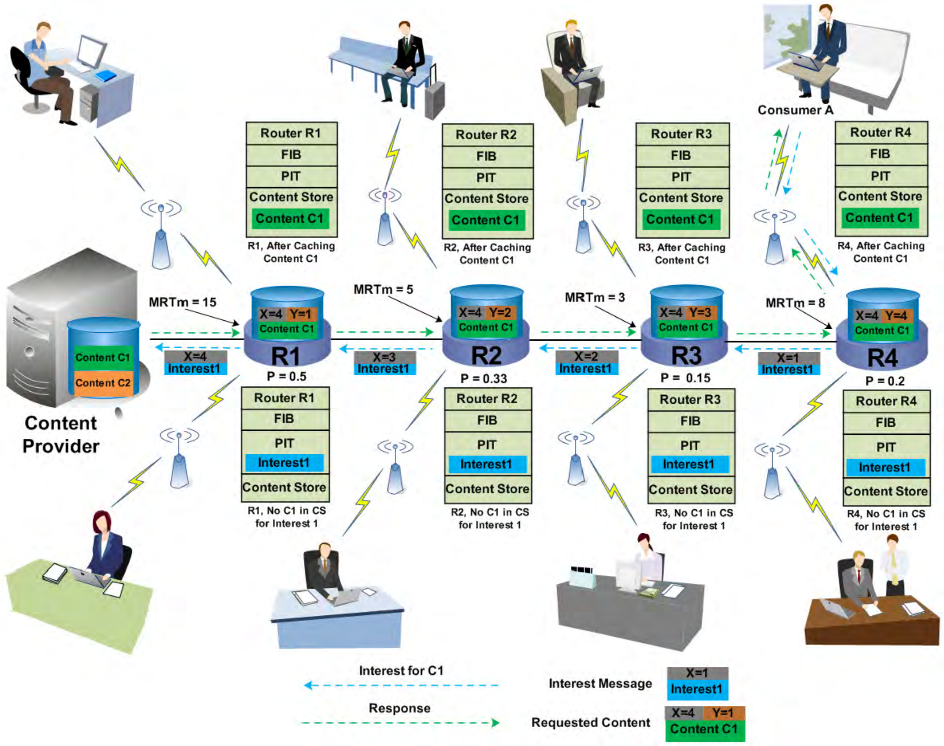


FIGURE 11. **Hop-based probabilistic caching.**

hop-count and is symbolized by α, where α is used as a constant integer to calculate the cache capacity of the whole network. The CacheWeight$_{\text{MRT}}$ factor is required to cache content for the particular time, which is dependent on two variables: mean residence time (MRTm) and expected mean residence time (MRT$_{\text{exp}}$) of content. The objective of HPC is illustrated in Fig. 11. Assume that each router has MRT$_{\text{exp}}$ $=$ 5 sec, $\alpha = 1$, and the values of the y parameter alongside the routers R1, R2, R3, and R4 are (1, 2, 3, and 4) in the company of MRT$_{\text{m}}$ $=$ (8, 3, 5, 15) sec. Subsequently, CacheWeighty will be CacheWeighty $=$ (0.5, 0.33, 0.15, 0.2), and the probability to store the content will be $p = (0.5, 0.33, 0.15, 0.2)$, as given in Fig. 11. Based on this scenario, it can be concluded that when the provider replies to a consumer Interest, the probability will increase as the distance decreases from the consumer. As a result, the HPC pushes the content gradually toward consumers [77]. HPC was developed to enhance the performance of edge caching by pushing the content near consumers and caching the content for a particular time. HPC introduced a new factor, (MRT), that is used to specify the time to cache content along the path, which increases resource usage. However, mean values do not establish any metrics in that case if the distribution functions of the content is neither normal nor uniform.

4) ROXCACHE

The ProXCache caching strategy has been derived recently from dynamic probabilistic caching to improve content redundancy and path redundancy. It implemented through a hypergraph, ProXCache strategy uses a specific header address combined with an Interest message as the consumer Interest for some content [18]. Based on the proposed caching, the dissemination of consumer Interest packets a particular cache time interval, path computation, and hop distance are required. When a cache hit is recorded, the interested data content is thrown to the consumer, and the proxy location is attained by the publisher to cache a copy of interested content along the consumer's path as an augmentation of Leave Copy Down (LCD) (see Fig. 12). ProXCache is different from the previous mechanisms due to the constant transmission of data delivered to consumers using cache time window, path distance, and hop distance. After that, the interested contents are cached probabilistically at the mid-routers of the consumer's path through a centrality computation algorithm. Therefore, transmitted contents are cached at the proxy node and moved toward the consumer. Through PIT records, this operation is performed at the intermediate node. In continuous Interests and network traffic, proxy positions, cache time windows, and path capacity of the network need to be obtained for each Interest. This caching mechanism was created to mitigate content redundancy and path redundancy; however in short stretch paths, it actually substantiates content redundancy. In addition, it increases the communication overhead because a number of variables need to be calculated during each transmission of Interest messages and interested content. ProXCache maximizes resource usage (caching) as millions of transmissions are required in a small time interval. Fig. 12 demonstrates the operation of the ProXCache caching

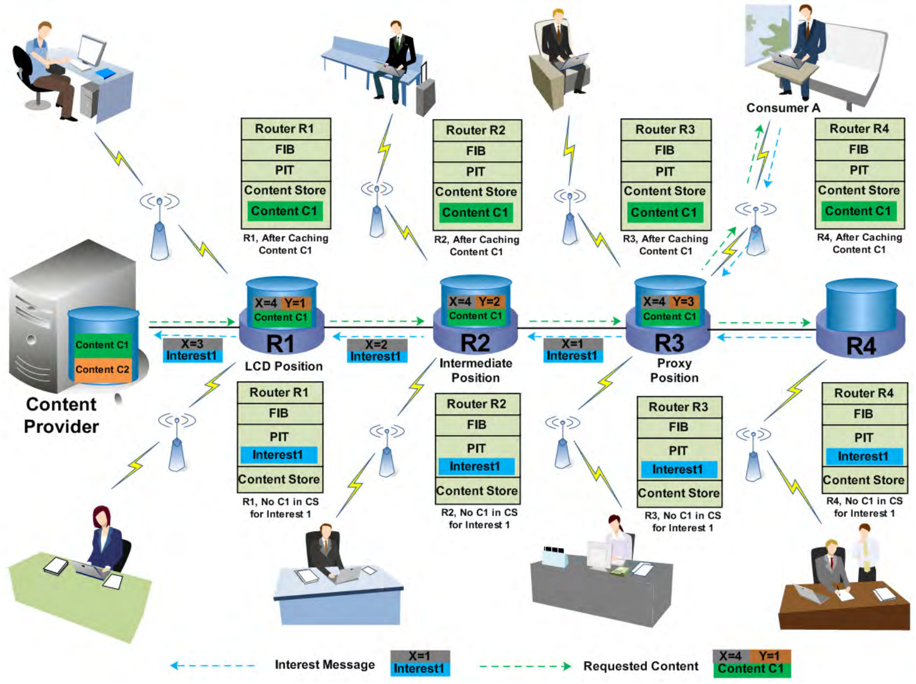


FIGURE 12. ProXCache.

strategy which clearly indicates the content redundancy along the short delivery path when Consumer A sends Interests messages. Consequently, the interested content C1 is cached at all the routers along the short stretch path by routers R1, R2, and R3. Moreover, ProXCache needs a number of parameters for the content dissemination which increases costs in term of resources and energy.

V. EVALUATION ENVIRONMENT

To compare and evaluate the performance of the discuss caching strategies, authors used a caching-based simulator called SocialCCNSim [23]. The performances were compared in terms of cache hit ratio, content diversity, stretch, and content redundancy of probabilistic caching strategies. For content replacement operations, most of the CCN caching strategies adopt the Least Recently Used (LRU) replacement policy [78]. LRU policy is the optimal content replacement policy due to its better performance in terms of overhead and complexity. Due to CCN's cache-driven architecture, various algorithms and strategies have been designed to choose from based on the set of rules regarding which content to cache from the varieties of popular content, content-defined names, and topological content information, among others. For the purposes of this study, the research will caching sets of content of four categories based on their popularity defined through Zipf's distribution [79]: web content, file sharing, user-generated content (UGC), and VoD [80]. The present study used Zipf 1.2 for VoD content because it shows high traffic production due to consumer interest in media-driven content.

TABLE 1. Simulation description values.

Parameter	Value/Description
Simulation Time	1 day
Chunk Size	10MB
Cache Size	1GB-10GB
Catalog Size	10^6
Alpha Value	1.2
Topology	Abilene
Replacement Policy	LRU
Simulator	SocialCCNSim
Traffic Source	SONETOR (Facebook [81])

A. CACHE HIT RATIO

The cache hit ratio presents the quantity of the average of existing content hits (found) as the Interests are sent [135]. Cache-hits can be defined [135], [146] as

$$CacheHitRatio = \frac{\sum_{n=1}^{n} Hit_n}{\sum_{n=1}^{n} (Hit + Miss)} \tag{10}$$

Numerous networking and database research studies have prioritized the cache hit ratio as method to improve network performance. For a specific strategy, the cache hit ratio can be obtained by calculating the total number of interests and misses for nodes. For the benefit of comparison, the significance of the cache hit ratio is pictorially represented in Fig. 13(a, b). As cache size expands, the performance of the network improves noticeably, excluding the EPC because of its probabilistic nature to cache content within all routers. Fig. 13 illustrates the consequences that occur when the Abilene topology is used while $\alpha = 1.2$. The x-axis is split into 10 equal sections that represent the cache size starting from 100 to 1,000. Fig. 13(a) shows the cache hit ratios of FIX strategies. EPC shows the same performance at both cache sizes because it has a similar amount of content replication for the different cache sizes. The PBPC performs better than EPC because it prioritizes the desired content, which increases the chances to cache content near consumers, heightening the hit ratio. However, RPC performs better than the other two compared strategies because of its random caching structure, which provides a better way to make caches useful. Fig. 13(b) shows the results of dynamic probabilistic caching on the cache hit ratio. These caching strategies exhibit the same performance, except that ProXCache performs better than the other dynamic caching strategies because ProbCache, ProbCache+, and HPC have homogeneous content replications that decrease the chances to improve diverse content caching. ProXCache provides a new way to decrease identical replications of content, which raises the possibility to accommodate heterogeneous content. When we expand the cache size, the performance seems to be similar to the small cache size, as shown in Fig. 13(b).

B. CONTENT REDUNDANCY

Content redundancy is a significant metric to assess heterogeneous data in CCN dissemination. EPC produces huge number of duplications that affect the whole performance and increase congestion. Content redundancy can be calculated with the following equation:

$$\text{Redundancy} = \sum\nolimits_{i=1}^{n} Rc_i \tag{11}$$

In this equation, Rc_i shows the redundancy at a specific position. The result in Fig. 13(c, d) illustrates that the noticeable cause of similar frequent caching operations is made by EPC due to its high-level usage of caches, which reinforces content redundancy. PBPC reduces the redundant content at multiple locations due to its nature of caching content near consumers. RPC performs better than EPC and PBPC because of its random nature of caching content, which increases the availability of diverse caching operations and reduces the replication of similar contents. Except for EPC, both PBPC and RPC show lower redundancy within large cache sizes. On the other hand, dynamic probabilistic caching strategies perform slightly differently because all the dynamic strategies have the same caching structure. However, ProXCache performs better than other compared strategies because its structure is changed which provides an improved method to enhance the caching mechanism that decreases content redundancy. Conversely, ProXCache shows a low level of homogeneity due to its proxy positions.

C. CONTENT DIVERSITY

The ratios of different sets of content accumulate in network caches. Different types of content need to be stored in a cache to increase content diversity. Diversity can be calculated as

$$Diversity = \frac{\bigcup_{v=1}^{V} X_v}{\sum\limits_{v=1}^{N} C_v} \tag{12}$$

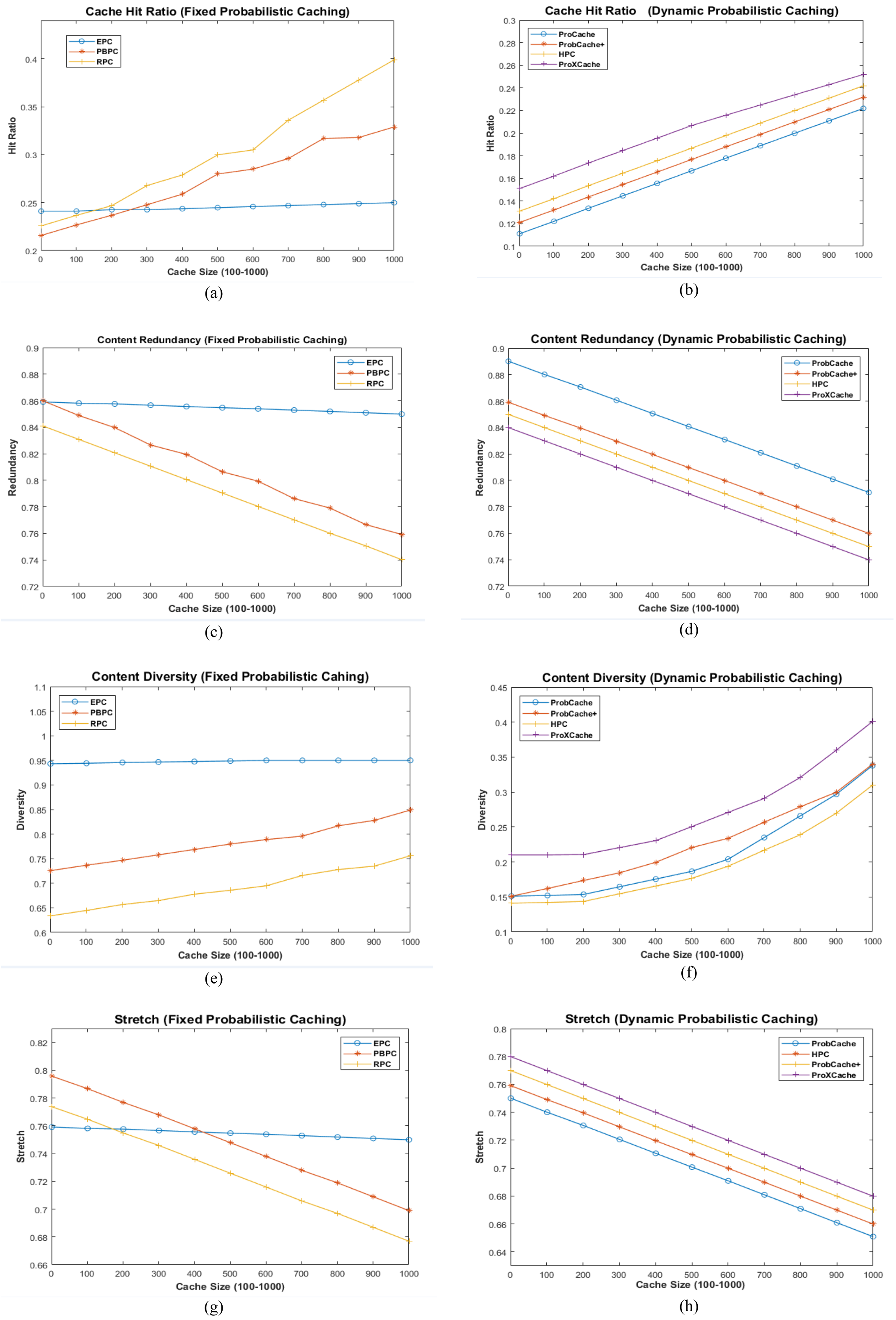
where V is the total number of nodes within the network, $\bigcup_{v=1}^{V} v$ represents the unique content X at the n routers, and $\sum_{v=1}^{N} C_v$ shows the cache of all the Cv routers. The intention of this probabilistic caching scenario is to minimize the high amount of homogeneous content replication in caches to accommodate the newly interested miscellaneous content. Moreover, the replication of similar content increases the replication of content that causes network congestion and high traffic. In addition, it diminishes the accommodation for heterogeneity [82]. On the other hand, diversity reduces the gigantic amount of content replicas that heighten the amount of heterogeneous content. Moreover, diversity identifies the distinctive content caching on discriminatory nodes and provides the ratio between the amount of unique and similar content accommodated at the same cache location. The FIX strategies perform better than the dynamic probabilistic strategies because dynamic strategies need to perform extra functionalities to make caching decisions, which increases communication overhead. Therefore, FIX seems to be better in performance, as shown in Fig. 13(e, f).

D. STRETCH

Stretch can be defined as the number of hops (routers) needed to be covered by a consumer's Interest between source (where the hit occurs) and consumer. When a consumer sends Interests for some content, the Interest needs to be covered by a number of hops to reach the source for the required content. Sometimes, the Interest finds a copy of the required content from any of the routers that appear on the path; otherwise, the Interest has to go to the server, which increases the average cost. Therefore, these caching strategies try to reduce the maximum number of hops between source and consumer. The stretch can be calculated as

$$\text{Stretch} = \frac{\sum_{i=1}^{R} Hop - traveled}{\sum_{i=1}^{R} Total - Hop} \tag{13}$$

where $\sum_{i=1}^{\boldsymbol{R}} \boldsymbol{Hop - traveled}$ represents the number of hops between consumer and source (where the cache hits occur), $\sum_{\boldsymbol{i}=1}^{|\boldsymbol{R}|} \boldsymbol{Total - Hop}$ is the total number of hops (between consumer and the server), and R is the total number of Interests issued by the consumer. FIX strategies show lower stretch than dynamic strategies because dynamic strategies support the similar content's replications, which decrease the amount of free cache space to accommodate new content. All consumer Interests need to be forwarded to the main sources, which enlarges the distance between consumer and source. Therefore, FIX performs better than dynamic probabilistic strategies, as demonstrated in Fig. 13 (g, h).



(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 13. **Simulation results on fixed versus dynamic probabilistic caching strategies. (a) Cache hit ratio (Abilene). (b) Cache hit ratio (Abilene). (c) Content redundancy (Abilene). (d) Content redundancy (Abilene). (e) Content diversity (Abilene). (f) Content diversity (Abilene). (g) Stretch (Abilene). (h) Stretch (Abilene).**

VI. ANALYSIS AND DISCUSSION

In Table 2, a number of CCN probabilistic caching strategies have been developed and examined in testing environment through different criteria. The EPC CCN caching mechanism was proposed to increase the data availability near the consumers [83]. This mechanism supports the redundancy

TABLE 2. Contribution and challenges of CCN probabilistic caching strategies.

Caching Mechanisms	Aim/Target	Caching Model	Gain	Challenges
Everywhere Probabilistic Caching	• Reduce the content redundancy • High cache hit	Autonomous	• Simplicity • Low overhead	• No position distinction • Favors low-stretch delivery paths Cache redundancy on short-stretch
Priority-based Probabilistic Caching	• Reduce the distance from the source	Autonomous	• Average cost	• No position distinction • Cache redundancy on short-stretch
Random-Probabilistic Caching	• Randomly cache content	Autonomous	• Simplicity Low overhead • Reduce delay	• Unwanted cache • No content distinction • No position distinction • Unpredictable random nature
ProbCache	• Fair allocation of capacity resources on a delivery path • among contents	cooperative	• Fair allocation of capacity resources	• Update of content request/content reply packets • Unfair allocation of cache capacity • (unfulfilled goal) • Arbitrary definition of parameters
ProbCache+	• Fair allocation of capacity resources on a delivery path • among contents	cooperative	• Fair allocation of capacity resources	• No content distinction • Update of content request/content reply packets (computational overhead) • Arbitrary definition of parameters
Hop-based Probabilistic Caching (HPC)	• Progressively cache content towards consumers for a specific time-interval	cooperative	• Fast content dissemination	• Global network knowledge • Lower caching probability towards • consumers (unfulfilled goal) • Inaccurate MRTm estimations
ProXCache	• Cache at Proxy Nodes near to the according to cache algorithm	Cooperative	• Reduce Bandwidth and delay • Low overhead • Low redundancy	• Update content information • Resource consumption

that increases the utilization of resources unnecessarily that caused the large number of caching and eviction operations and reduce the cache space for incoming content, due to which the stretch is maximize because new Interests need to forward to the main source due to less priority to cache most popular contents nearest the interested consumers. Moreover, it supports the content homogeneity that decreases the amount of diverse contents within the cache of the node and consequently the utilization of resources is exploiting. In addition, the simulation results also proves these issues in Fig. 13(a) and it shows the same effects for different caches sizes [58].

The PBPC performs well than EPC because of its content caching structure. It decreases the distance for the subsequent Interests that results high performance in terms of caching gain, stretch, diversity and redundancy, but some issues are still exists in this caching structure, For example, the content will cache at all the node alongside the data downloading path when the path length is very small, which maximize the redundancy, usage of resources (i.e., cache) and it will reduce diversity as well as cache hit when the cache is overflow. The RPC executed the better results to achieve high cache hit, diversity and low stretch. Moreover, it also performs better in terms stretch because its caching operation is done according to free cache space. Suppose, the interested content will cache near the consumer because it has more chance to get free cache at the edge nodes its random nature of caching operations. Therefore, it reduces the stretch and diversity and heightens the hit rate for the subsequent Interests.

ProbCache provides fair allocation of resources along the delivery path amongst disseminated contents but it has no any content distinction, updating of Interest and content reply packets (computational overhead) arbitrary description of parameters [58]. This strategy does not succeeds to provide any compensation which proves that it does not get done its goals [60]. Both fixed and dynamic probabilistic caching endeavor to diminish the content redundancy but the redundancy still exist. The dynamic probabilistic strategies established a new attribute Ttw which is used to determine a particular time interval to cache content alongside the data path but it reinforce the content eviction rate. Fixed probabilistic caching strategies have low overhead but there is no any cooperation amongst the nodes which maximize the redundancy and enlarge the consumption of the resources. Random probabilistic cache scheme introduced simplicity and low overhead but it does not have any content as well as position distinction and it deliver unpredictable nature [65], [66]. ProbCache and ProbCache+ aim to tackle the problems related to the inefficient usage of the resources in a different way. They share the resources (cache) fairly among the contents but these strategies seem to be unable to achieve their goals completely [31].

Both caching strategies introduce a number of parameters that have no any content distinction and need to set these parameters arbitrarily. Both strategies additionally need to update the consumer Interests and contents in the form of TSI and TSB that increase the computational cost and memory consumption within the all routers. Moreover, these caching strategies introduce the Ttw by using a special variable that also increases the computational cost. In addition, these strategies have no any distinction about the cached contents and there is no criterion to handle the frequently Interested contents [77]. Both strategies perform slightly different in simulation environment and they did not improve the overall caching performance as compare to the fixed probabilistic strategies. The hope based caching was developed to enhance the performance of edge caching by pushing the contents near to the consumers and cache it for a particular time. HPC introduced a new factor MRT that is used to specify the time to cache a content along the path which increase the usage of the resources. Although, mean values do not establish any metric in that case if the distribution function of the contents is neither normal nor uniform [70]. It also keeps the high level of content redundancy that increases the amount of homogeneous caching operation that results the diversity is reduce. Moreover, the similar caching operations reduce the hit rate due to the small amount of cache space because it does not accommodate the diverse number of content. ProXCache caching scheme has benefits related to the useful information and this mechanism works like a server that prevents the network from harm data [18]. It reduces the content redundancy and increase the content diversity through implementation of proxy positions. Moreover, this mechanism is capable to reduce the caching and eviction operations that minimize the resource consumption and increase the usability of popular contents.

Similarly, in simulation environment it implements decent results in terms to improve the hit rate, diversity, stretch and content redundancy. But, there is no any criterion to handle the redundant caching operation when the stretch path is very small as shown in Figure 12. Regarding this, it cannot stop the homogeneous caching operation, which caused the high redundancy and stretch. Moreover, it will reduce the free cache space to putt the diverse contents closer to the consumers to improve the cache hit ratio.

VII. CONCLUSION

As the IP network paradigm progressively moves from host-centric location based communication to receiver-driven content reclamation, a number of Content Centric Networking architectures have newly been projected. These architectures are going to make available aboriginal architectural support for extremely proficient and scalable content reclamation and to resolve the traffic explosion problem. In CCN network architecture, transparent and ubiquitous cache is an essential building block that promises efficient content retrieval. To summarize, CCN is a comparatively innovative research area for the Internet communications. In this paper, we first give some details about CCN architecture and its caching module after that several caching mechanisms are investigated. Moreover, CCN caching strategies are surveyed on the basis of basic caching performance metric such as; content redundancy, content diversity, stretch and cache hit ratio. In this survey, a number of probabilistic caching mechanisms are illustrated with graphical representation (architectures and simulations) as well as their advantages and disadvantages. All of them emphasize to perform better cache utilization in term of content redundancy and availability of desired data. However, there are several issues still exit in all mentioned probabilistic caching strategies. As a result, we can say that the research on CCN caching is still in its premature stage. There are a variety of technical and theoretical issues to be addressed.

REFERENCES

[1] J. Pan, S. Paul, and R. Jain, "A survey of the research on future Internet architectures," *IEEE Commun. Mag.*, vol. 49, no. 7, pp. 26–36, Jul. 2011.

[2] L. Linguaglossa, "Two challenges of software networking: Name-based forwarding and table verification," École doctorale, Sciences Mathématiques de Paris Centre Laboratoire, IRIF en collaboration avec, Nokia Bell Labs, INRIA, 2016.

[3] B. Banerjee, A. Kulkarni, and A. Seetharam, "Greedy caching: An optimized content placement strategy for information-centric networks," *Comput. Netw.*, vol. 140, pp. 78–91, Jun. 2018.

[4] O. Vermesan *et al.*, "Internet of Things strategic research roadmap," in *Internet of Things: Global Technological and Societal Trends*, vol. 1. 2011, pp. 9–52.

[5] A. Merentitis *et al.*, "Virtualization as a driver for the evolution of the Internet of Things: Remaining challenges and opportunities towards smart cities," *Int. J. Adv. Internet Technol.*, vol. 7, nos. 1–2, pp. 86–160, 2014.

[6] M. Garmehi, M. Analoui, M. Pathan, and R. Buyya, "An economic mechanism for request routing and resource allocation in hybrid CDN–P2P networks," *Int. J. Netw. Manage.*, vol. 25, no. 6, pp. 375–393, 2015.

[7] A. X. Liu, C. R. Meiners, and E. Torng, "TCAM Razor: A systematic approach towards minimizing packet classifiers in TCAMs," *IEEE/ACM Trans. Netw.*, vol. 18, no. 2, pp. 490–500, Apr. 2010.

[8] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. 5th Int. Conf. Emerg. Netw. Exp. Technol.*, 2009, pp. 1–12.

[9] G. Xylomenos *et al.*, "A survey of information-centric networking research," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1024–1049, 2nd Quart., 2014.

[10] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 26–36, Jul. 2012.

[11] L. Veltri, G. Morabito, S. Salsano, N. Blefari-Melazzi, and A. Detti, "Supporting information-centric functionality in software defined networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 6645–6650.

[12] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, "Network of Information (NetInf)—An information-centric networking architecture," *Comput. Commun.*, vol. 36, no. 7, pp. 721–735, 2013.

[13] A. V. Vasilakos, Z. Li, G. Simon, and W. You, "Information centric network: Research challenges and opportunities," *J. Netw. Comput. Appl.*, vol. 52, pp. 1–10, Jun. 2015.

[14] D. Grewe, M. Wagner, and H. Frey, "A domain-specific comparison of information-centric networking architectures for connected vehicles," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2372–2388, 3rd Quart., 2018.

[15] L. A. Sánchez, "To catch a wave: The Beach Boys and rock historiography," Ph.D. dissertation, School Comput., Univ. Utara Malaysia, Changlun, Malaysia, 2012.

[16] T. Sprinkle, *Screw the Valley: A Coast-to-Coast Tour of America's New Tech Startup Culture*. New York, NY, USA: BenBella Books, Inc., 2015.

[17] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A survey of green information-centric networking: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1455–1472, 3rd Quart., 2015.

[18] I. Abdullahi, S. Arif, and S. Hassan, "Cache-less redundancy using hypergraph in information-centric network," *Adv. Sci. Lett.*, vol. 21, no. 11, pp. 3546–3549, 2015.

[19] I. Abdullahi, S. Arif, and S. Hassan, "Survey on caching approaches in information centric networking," *J. Netw. Comput. Appl.*, vol. 56, pp. 48–59, Oct. 2015.

[20] Q. N. Nguyen, M. Arifuzzaman, T. Miyamoto, and S. Takuro, "An optimal information centric networking model for the future green network," in *Proc. IEEE 12th Int. Symp. Auto. Decentralized Syst. (ISADS)*, Mar. 2015, pp. 272–277.

[21] K. Katsaros, G. Xylomenos, and G. C. Polyzos, "MultiCache: An overlay architecture for information-centric networking," *Comput. Netw.*, vol. 55, no. 4, pp. 936–947, 2011.

[22] I. Ud Din, "Flexpop: A popularity-based caching strategy for multimedia applications in information-centric networking," Ph.D. dissertation, Univ. Utara Malaysia, Changlun, Malaysia, 2016.

[23] 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. 2576, 2018.

[24] S. Arshad, M. A. Azam, M. H. Rehmani, and J. Loo. (2017). "Information-centric networking based caching and naming schemes for Internet of Things: A survey and future research directions." [Online]. Available: https://arxiv.org/abs/1710.03473

[25] H. Jin, D. Xu, C. Zhao, and D. Liang, "Information-centric mobile caching network frameworks and caching optimization: A survey," *EURASIP J. Wireless Commun. Netw.*, vol. 2017, no. 1, p. 33, Dec. 2017.

[26] S. Rahel, A. Jamali, and S. El Kafhali, "Energy-efficient on caching in named data networking: A survey," in *Proc. 3rd Int. Conf. Cloud Comput. Technol. Appl. (CloudTech)*, 2017, pp. 1–8.

[27] S. Leong, "Navigating the emerging futures in music education," *J. Music, Technol. Educ.*, vol. 4, pp. 233–243, Feb. 2012.

[28] Z. Li, Y. Liu, Y. Chen, Y. Xu, and K. Liu, "Performance analysis of a novel 5G architecture via content-centric networking," *Phys. Commun.*, vol. 25, pp. 328–331, Dec. 2017.

[29] G. Panwar, R. Tourani, T. Mick, A. Mtibaa, and S. Misra, "DICE: Dynamic multi-RAT selection in the ICN-enabled wireless edge," in *Proc. Workshop Mobility Evolving Internet Archit.*, 2017, pp. 31–36.

[30] M. Alajlan and A. Belghith, "Supporting seamless mobility for real-time applications in named data networking," *Procedia Comput. Sci.*, vol. 110, pp. 62–69, 2017, doi: 10.1016/j.procs.2017.06.117.

[31] Y. Zhu, Z. Mi, and W. Wang, "A probability caching decision policy with evicted copy up in content centric networking," *J. Internet Technol.*, vol. 18, pp. 33–43, Jan. 2015.

[32] J. V. Torres, I. D. Alvarenga, R. Boutaba, and O. C. M. B. Duarte, "An autonomous and efficient controller-based routing scheme for networking named-data mobility," *Comput. Commun.*, vol. 103, pp. 94–103, May 2017.

[33] I. Solis, M. Mosko, G. Scott, and A. Walendowski, "CCNx 1.0 tutorial: Theory and practice," in *Proc. 2nd Int. Conf. Inf.-Centric Netw.*, 2015, pp. 1–2.

[34] H. Huang, L. Rui, W. Zheng, D. Niu, X. Qiu, and S. Shao, "General, practical and accurate models for the performance analysis of multi-cache systems," *IEICE Trans. Commun.*, vol. E99-B, no. 12, pp. 2559–2573, 2016.

[35] 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.

[36] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[37] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A case for stateful forwarding plane," *Comput. Commun.*, vol. 36, no. 7, pp. 779–791, Jul. 2013.

[38] J. Daubert, M. Fischer, T. Grube, S. Schiffner, P. Kikiras, and M. Mühlhäuser, "AnonPubSub: Anonymous publish-subscribe overlays," *Comput. Commun.*, vol. 76, pp. 42–53, Feb. 2016.

[39] A. Rayes and S. Salam, "Internet of Things (IoT) overview," in *Internet of Things From Hype to Reality*. Springer, 2017, pp. 1–34.

[40] A. Herdrich *et al.*, "Cache QoS: From concept to reality in the Intel Xeon processor E5-2600 v3 product family," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Mar. 2016, pp. 657–668.

[41] Y. Wang, J. Bi, and K. Zhang, "Design and implementation of a software-defined mobility architecture for IP networks," *Mobile Netw. Appl.*, vol. 20, no. 1, pp. 40–52, 2015.

[42] M. Lee, J. Song, K. Cho, S. Pack, J. Kangasharju, and Y. Choi, "Content discovery for information-centric networking," *Comput. Netw.*, vol. 83, pp. 1–14, 2015.

[43] E. G. AbdAllah, M. Zulkernine, and H. S. Hassanein, "DADI: Defending against distributed denial of service in information-centric networking routing and caching," *Secur. Privacy*, vol. 1, no. 2, p. e16, 2018.

[44] A. Araldo, D. Rossi, and F. Martignon, "Cost-aware caching: Caching more (costly items) for less (ISPs operational expenditures)," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 5, pp. 1316–1330, May 2016.

[45] A. Ioannou and S. Weber, "Towards on-path caching alternatives in information-centric networks," in *Proc. IEEE 39th Conf. Local Comput. Netw. (LCN)*, Sep. 2014, pp. 362–365.

[46] A. Seetharam, "On caching and routing in information-centric networks," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 204–209, Mar. 2018.

[47] L. Rui, H. Guo, H. Huang, R. Shi, and X. Qiu, "MUPF: Multiple unicast path forwarding in content-centric VANETs," *Ad Hoc Netw.*, vol. 58, pp. 1–292, Apr. 2017.

[48] B. Ohlman, E. Davies, S. Spirou, K. Pentikousis, and G. Boggia, *Information-Centric Networking: Evaluation and Security Considerations*, document RFC 7945, 2016.

[49] C. Ghali, G. Tsudik, and C. A. Wood, "When encryption is not enough: Privacy attacks in content-centric networking," in *Proc. 4th ACM Conf. Inf.-Centric Netw.*, 2017, pp. 1–10.

[50] L. Wang, S. Bayhan, J. Ott, J. Kangasharju, and J. Crowcroft, "Understanding scoped-flooding for content discovery and caching in content networks," *IEEE J. Sel. Areas Commun.*, to be published.

[51] L. Ding, J. Wang, Y. Sheng, and L. Wang, "An efficient ranking-matched caching strategy for information centric networking," *Int. J. Innov. Comput. Inf. Control*, vol. 14, no. 2, pp. 519–535, 2018.

[52] V. Pacifici and G. Dan, "Paper D: Distributed algorithms for content allocation in inter-connected content distribution networks," in *Proc. 29th Conf. Inf. Commun. (INFOCOM)*, San Diego, CA, USA, Mar. 2010, pp. 1478–1286.

[53] S. P. Narayanan, Y. S. Nam, A. Sivakumar, B. Chandrasekaran, B. Maggs, and S. Rao, "Reducing latency through page-aware management of web objects by content delivery networks," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 44, no. 1, pp. 89–100, 2016.

[54] G. Rossini and D. Rossi, "Evaluating CCN multi-path interest forwarding strategies," *Comput. Commun.*, vol. 36, no. 7, pp. 771–778, 2013.

[55] H. Nakajima, T. Tsutsui, H. Urabayashi, M. Yamamoto, E. Rosensweig, and J. F. Kurose, "Performance evaluation of partial deployment of an in-network cache location guide scheme, breadcrumbs," *IEICE Trans. Commun.*, vol. E99-B, no. 1, pp. 157–166, 2016.

[56] D. Saucez *et al.*, *Information-Centric Networking (ICN) Research Challenges*, document RFC 7927, 2016.

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

[58] Y. Li, T. Zhang, X. Xu, Z. Zeng, and Y. Liu, "Content popularity and node level matched based probability caching for content centric networks," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Jul. 2016, pp. 1–6.

[59] R. S. Abbas, "Performance evaluation of caching placement algorithms in named data network for video on demand service," M.S. thesis, School Comput., Univ. Utara Malaysia, Changlun, Malaysia, 2016.

[60] S. Arianfar, P. Nikander, and J. Ott, "On content-centric router design and implications," in *Proc. Re-Architecting Internet Workshop*, 2010, p. 5.

[61] 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.

[62] W. Ding, Z. Yan, and R. H. Deng, "A survey on future Internet security architectures," *IEEE Access*, vol. 4, pp. 4374–4393, 2016.

[63] G. Carofiglio, L. Mekinda, and L. Muscariello, "LAC: Introducing latency-aware caching in information-centric networks," in *Proc. IEEE 40th Conf. Local Comput. Netw. (LCN)*, Oct. 2015, pp. 422–425.

[64] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Scalable mobile backhauling via information-centric networking," in *Proc. IEEE Int. Workshop Local Metrop. Area Netw. (LANMAN)*, Apr. 2015, pp. 1–6.

[65] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1473–1499, 3rd Quart., 2015.

[66] Q. Chen, R. Xie, F. R. Yu, J. Liu, T. Huang, and Y. Liu, "Transport control strategies in named data networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2052–2083, 3rd Quart., 2016.

[67] J. Du, Y. He, D. Wan, and J. Xiao, "Provisioning a cluster of distributed computing platform based on placement strategy," U.S. Patent 9 268 590 B2, Feb. 23, 2016.

[68] C. Y. Chang and M. S. Chang, "A hybrid coordination approach of in-network caching for named data networking," *Int. J. Future Gener. Commun. Netw.*, vol. 9, no. 4, pp. 285–300, 2016.

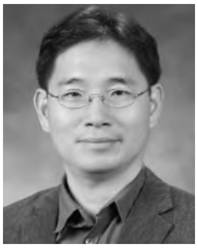
[69] G. Zhang, Y. Li, and T. Lin, "Caching in information centric networking: A survey," *Comput. Netw.*, vol. 57, pp. 3128–3141, Nov. 2013.

[70] I. Psaras, W. K. Chai, and G. Pavlou, "In-network cache management and resource allocation for information-centric networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 2920–2931, Nov. 2014.
[71] N. Abani, T. Braun, and M. Gerla, "Proactive caching with mobility prediction under uncertainty in information-centric networks," in *Proc. 4th ACM Conf. Inf.-Centric Netw.*, 2017, pp. 88–97.
[72] M. Namba, "A recursive estimation of network state for improving probabilistic caching," in *Proc. 7th Int. Conf. Soft Comput. Pattern Recognit. (SoCPaR)*, Nov. 2015, pp. 270–273.
[73] O. Ascigil, V. Sourlas, I. Psaras, and G. Pavlou, "A native content discovery mechanism for the information-centric networks," in *Proc. 4th ACM Conf. Inf.-Centric Netw.*, 2017, pp. 145–155.
[74] W. Li, "Popularity-driven caching strategy for dynamic adaptive streaming over information-centric networks," M.S. thesis, School Comput., Queen's Univ., Kingston, ON, Canada, 2015.
[75] H. Wu, J. Li, T. Pan, and B. Liu, "A novel caching scheme for the backbone of named data networking," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2013, pp. 3634–3638.
[76] Y. Sun, T. Zhang, R. Wang, W. Feng, and P. Chen, "Topology potential based probability caching strategy for content centric ad hoc networks," *J. Residuals Sci. Technol.*, vol. 13, no. 6, pp. 104.1–104.6, 2016.
[77] H. Wu, J. Li, and J. Zhi, "MBP: A max-benefit probability-based caching strategy in information-centric networking," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 5646–5651.
[78] L. Saino, I. Psaras, and G. Pavlou, "Icarus: A caching simulator for information centric networking (ICN)," in *Proc. 7th Int. ICST Conf. Simulation Tools Techn.*, 2014, pp. 66–75.
[79] A. Lareida, R. Pernischova, B. B. Rodrigues, and B. Stiller, "Abstracting .torrent content consumption into two-mode graphs and their projection to content networks (ConNet)," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, May 2017, pp. 151–159.
[80] R. Jmal and L. C. Fourati, "An OpenFlow architecture for managing content-centric-network (OFAM-CCN) based on popularity caching strategy," *Comput. Stand. Inter.*, vol. 51, pp. 22–29, Mar. 2017.
[81] I. Cantador, P. L. Brusilovsky, and T. Kuflik, "Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011)," in *Proc. 5th ACM Conf. Recommender Syst. (RecSys)*, 2011, pp. 387–388.
[82] İ. Gülçın, M. Oktay, E. Kıreçcı, and Ö. İ. Küfrevıoğlua, "Screening of antioxidant and antimicrobial activities of anise (*Pimpinella anisum* L.) seed extracts," *Food Chem.*, vol. 83, no. 3, pp. 371–382, 2003.
[83] H. Tian, Y. Otsuka, M. Mohri, Y. Shiraishi, and M. Morii, "Leveraging in-network caching in vehicular network for content distribution," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 6, p. 8972950, 2016.

MUHAMMAD ALI NAEEM received the M.Sc. degree in computer science from the Comsats institute of Information Technology, Pakistan He is currently pursuing the Ph.D. degree in networking (Information Centric Networking). He is also a Ph.D. Scholar at the InterNetWorks Research Lab, School of Computing, College of Arts and Sciences, Universiti Utara Malaysia.

SHAHRUDIN AWANG NOR received the B.Eng. degree (Hons.) in electronics engineering from Universiti Sains Malaysia, the M.Sc. degree in computer and information networks from the University of Essex, U.K., and the Ph.D. degree in computer networks from Universiti Utara Malaysia. He is currently a Senior Lecturer and the Head of platform at the InterNetWorks Research Lab, School of Computing, College of Arts and Sciences, Universiti Utara Malaysia. His research interests include computer network and security, communication and network performance, network transport protocol, and computer organization.

SUHAIDI HASSAN (S'01–M'03–SM'08) is currently a Professor of computing networks at the School of Computing, Universiti Utara Malaysia (UUM). He is the Founding Chair of the InterNetWorks Research Laboratory, UUM, which spearheads advanced research on Internet computing and protocol engineering. He is a Senior Member of the US-based IEEE, and actively involved in both IEEE Communications and IEEE Computer societies. He is the Internet Society Fellow to the Internet Engineering Task Force (IETF). In 2006, he was selected as a WKD Foundation (Switzerland)'s Young Scientist Fellow at the World Knowledge Dialogue, Crans-Montana, Switzerland. In 2006, he led a task force for the establishment of the ITU-UUM Asia Pacific Centre of Excellence for Rural ICT Development, a human resource development initiative of the Geneva-based International Telecommunication Union (ITU) which serves as the focal point for all rural ICT development initiatives across Asia Pacific region. In addition to presenting research papers at renown research conferences, he participates in international fora, such as ICANN meetings, Internet governance forums, the IETF meetings, and the IEEE meetings. He has authored and co-authored over 200 refereed technical publications, successfully supervised 20 Ph.D. scholars in the area of computer and communication networks, served as the reviewer and referee for refereed journals and conferences on computing networks and the examiner for over fifty doctoral and postgraduate scholars in his research areas.



BYUNG-SEO KIM (M'02–SM'17) received the B.S. degree in electrical engineering from Inha University, Inchon, South Korea, in 1998, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Florida in 2001 and 2004, respectively. His Ph.D. study was supervised by Dr. Y. Fang. From 1997 to 1999, he was with Motorola Korea Ltd., Paju, South Korea, as a Computer Integrated Manufacturing Engineer at the Advanced Technology Research and Development. From 2005 to 2007, he was with Motorola Inc., Schaumburg Illinois, as a Senior Software Engineer in networks and enterprises. His research focuses at Motorola Inc., were designing protocol and network architecture of wireless broadband mission critical communications. He is currently a Professor at the Department of Software and Communications Engineering, Hongik University, South Korea. His work has appeared in around 161 publications and 22 patents. His research interests include the design and development of efficient wireless/wired networks, including link adaptable/ cross-layer-based protocols, multi-protocol structures, wireless CCNs/NDNs, mobile edge computing, physical layer design for broadband PLC, and resource allocation algorithms for wireless networks. He was the Chairman of the Department from 2012 to 2014. He served as the General Chair for 3rd IWWCN 2017, and a TPC Member for the IEEE CAMAD 2018, the IEEE VTC 2014-Spring, EAI FUTURE2016, and ICGHIT 2016-2019 conferences. He was also served as a Member of the Sejong-city Construction Review Committee and the Ansan-city Design Advisory Board. He served as the Guest Editors of special issues of the *International Journal of Distributed Sensor Networks* (SAGE), the IEEE ACCESS, the *Journal of Ambient Intelligence*, *Sensors*, and *Humanized Computing*, and the *Journal of the Institute of Electronics and Information Engineers*. He is an Associate Editor of the IEEE ACCESS.

• • •