Caching in Information-Centric Networking: Strategies, Challenges, and Future Research Directions

Ikram Ud Din, *Student Member, IEEE*, Suhaidi Hassan, *Senior Member, IEEE*, Muhammad Khurram Khan, *Senior Member, IEEE*, Mohsen Guizani, *Fellow, IEEE*, Osman Ghazali, *Member, IEEE*, and Adib Habbal, *Senior Member, IEEE*

***Abstract*—Information-Centric Networking (ICN) is an appealing architecture that has received a remarkable interest from the research community thanks to its friendly structure. Several projects have proposed innovative ICN models to cope with the Internet practice, which moves from host-centrism to receiver-driven communication. A worth mentioning component of these novel models is in-network caching, which provides flexibility and pervasiveness for the upturn of swiftness in data distribution. Because of the rapid Internet traffic growth, cache deployment and content caching have been unanimously accepted as conspicuous ICN issues to be resolved. In this article, a survey of cache management strategies in ICN is presented along with their contributions and limitations, and their performance is evaluated in a simulation network environment with respect to cache hit, stretch ratio, and eviction operations. Some unresolved ICN caching challenges and directions for future research in this networking area are also discussed.**

***Index Terms*—ICN, future Internet, caching strategies, Internet architecture, off-path caching, on-path caching.**

I. Introduction

THE EXISTING Internet was intended to address the communication demands of a period when a network was required to share expensive and rare resources, such as long distance communication links, peripherals, and mainframe computers [1]. The main design rules of the Internet made it possible to connect new systems and allow a remarkable growth in its size. However, the inspiring growth of the Internet has offered ascend to new requirements from the design, for example, content dissemination and storage resources [2]. Moreover, due to its particular implementation and end-to-end approach, the current Internet architecture has many limitations. For example, a variety of add-on patches, such as Point-to-Point (P2P) overlays, Content Delivery Networks (CDNs), and Network Address Translation (NAT), that were not part of the original design, all violate, in different manners, various features of the initial Internet architecture [3], [4].

It has been recently perceived that information-centric usage of the Internet has gained popularity [5], which has been revealed by the majority associated with it [6]. This usage brings up a range of design challenges, several of which were not successfully handled by the existing architecture, such as tussle mediation through information governance and medium-independent information access [7].

Here, a question has been raised that whether the Internet needs a new clean-slate design method or we can keep "patching over patches" [9]. Therefore, a research group [6] has been formed to identify the impediments of the existing Internet and discuss the major objectives and requirements of the future Internet architecture. In this situation, Information-Centric Networking (ICN) has turned into a favorable nominee for the future Internet architecture.

The data tsunami was predicted in the mid of 1990s, which fortunately did not come, but due to exponential growth in the number of Internet users and interest for multimedia applications [10]–[12], the alarm of data tsunami is returning [13]. The Cisco Visual Networking Index [14] exhibits that the delivery of multimedia applications has become extremely popular on the Internet. Surprisingly by 2021, the global IP traffic will be 194 exabytes per month or 2.3 zettabytes per year. The widespread Internet multimedia traffic will reach 82 percent of all users Internet traffic, which was 70 percent in 2015. In addition, it is predicted that the number of communicating devices will be three times more than the world population by 2021 [15]. The aggregation of all kinds of videos, i.e., video on demand (VoD), Point to Point (P2P), and Television (TV) will be roughly 86 percent of the worldwide user traffic [16]. This evolution is more vividly portrayed in Figure 1 which represents the number of data and its type that is transferred in an Internet minute [8].

Figure 1 demonstrates a strong hold of how the present host-to-host Internet is overloaded with levels of popularity of information. As indicated by the most recent data traffic and

Manuscript received September 12, 2016; revised December 15, 2016, March 15, 2017, July 24, 2017, and November 22, 2017; accepted December 11, 2017. Date of publication December 27, 2017; date of current version May 22, 2018. This work is supported by the Ministry of Higher Education Malaysia with the Fundamental Research Grant Scheme under Grant Number FRGS-13258, and the Deanship of Scientific Research at King Saud University with the Research Group Number RGP-288. *(Corresponding author: Suhaidi Hassan.)*

I. U. Din, S. Hassan, O. Ghazali, and A. Habbal are with the InterNetWorks Research Laboratory, School of Computing, Universiti Utara Malaysia, Sintok 06010, Malaysia (e-mail: suhaidi@uum.edu.my).

M. K. Khan is with the Center of Excellence in Information Assurance, King Saud University, Riyadh 11653, Saudi Arabia.

M. Guizani is with the Department of Electrical and Computer Engineering, University of Idaho, Moscow, ID 83844 USA.

Digital Object Identifier 10.1109/COMST.2017.2787609

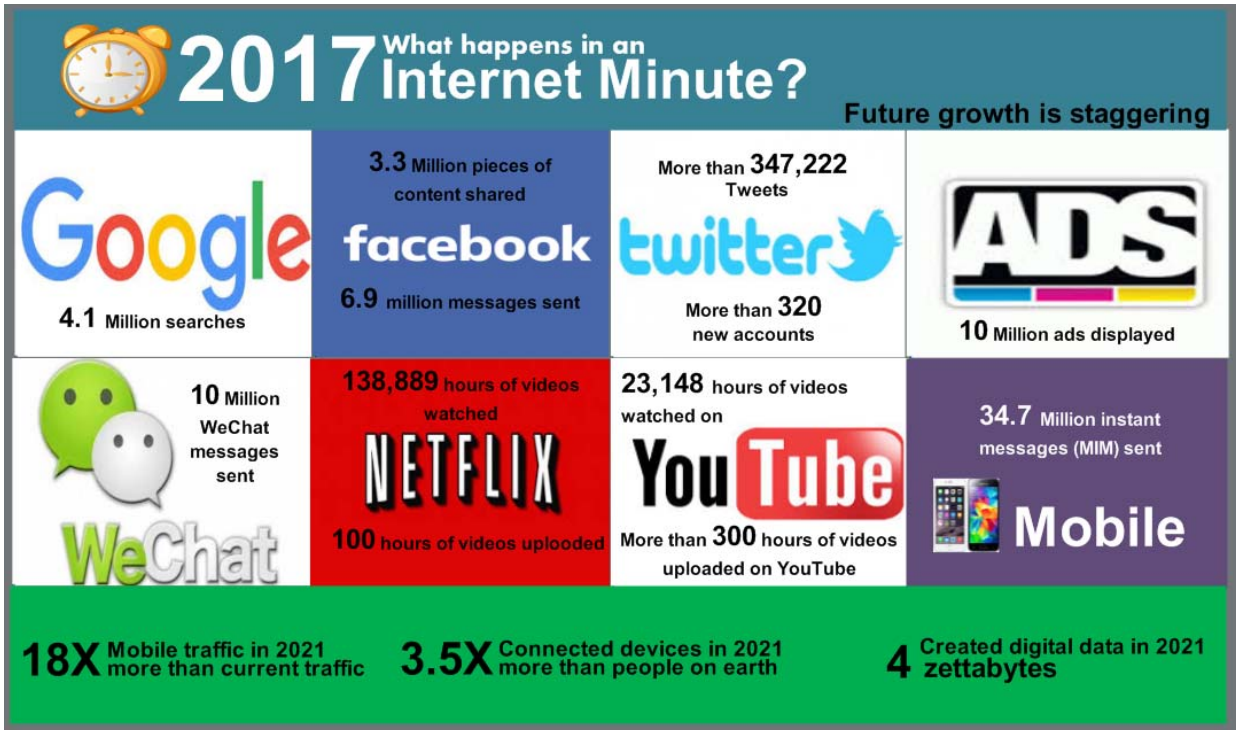


Fig. 1. Internet in a minute: Adapted from [8].

information sharing on the Internet in a minute [8], [14], [18], an approximated more than 34.7 million messages are sent over the Internet. Furthermore, mobile traffic will account for 30 percent of total IP traffic in 2021, which was recorded only 8 percent in 2015. Moreover, Google being utilized as one of the leading search engines, received a maximum number of hits, i.e., 4.1 million searches in an Internet minute. With this data calculation, where it is noticeable that maximum shared/accessed data is multimedia [19], [20], it is worthy to separate host-centrism of data from information-centric architecture. This may be easily manageable by flexibly placing caching nodes in the network through ICN concepts.

The rest of this article is organized as follows: We first introduce the design concept of ICN, then relevant ICN studies and its general idea, including its background, main components, and the workflow and architecture of the basic ICN approach. After defining the ICN architecture, we explain ICN in-network caching and then present the existing cache management strategies. We also summarize some research challenges and directions for future research in ICN cache management. Then we evaluate the performance of these strategies through simulations, and finally we conclude the paper.

II. Design Concept of ICN

The configuration of ICN is one of the notable after-effects of various global future Internet research activities. The current Internet architecture depends on end-to-end correspondence between hosts, which is called host-centric networking [21], [22]. Instead, the growing demand for efficient and highly scalable distribution of contents (for example, documents, videos, Web pages, or other types of information) has motivated the modification of architectures that concentrate on Named Data Objects (NDOs) [23]. The architectures of ICN can influence multiparty correspondence through replication, in-network caching, and interaction models such as publish-subscribe to achieve effective and consistent transmission of contents by providing a common platform for communication services [24]. The ICN approach is being investigated by various research projects, including the U.S. funded projects: Data Oriented Network Architecture (DONA) [25], Named Data Networking (NDN) [26]–[28], MobilityFirst [29], Content-Centric Networking (CCN) [30], and the European funded projects: Publish-Subscribe Internet Routing Paradigm (PSIRP) [31], 4WARD [32], [33], Publish-Subscribe Internet Technology (PURSUIT) [34], CONVERGENCE [35], Scalable & Adaptive Internet soLutions (SAIL) [36], and COntent Mediator architecture for content-aware nETworks (COMET) [37]. While the approaches of these projects differ concerning their specific design, they share some architectural properties, objectives, and suppositions [24].

Generally, the purpose is to create system models that are appropriate for content distribution and that better adapt to flash-crowd impacts, disruptions, and disconnections in the communication service [17]. Communication is driven by recipients *requesting* contents, while receivers publish the contents so that they are available to the senders.

In Figure 2, the network is divided into autonomous systems (AS), i.e., AS1 and AS2, respectively. A user (User1) in AS1, connected to node N3, requests content C, which was already cached at node N1. Thus, the request is satisfied by node N1, rather than downloading from the publisher (S3) having the

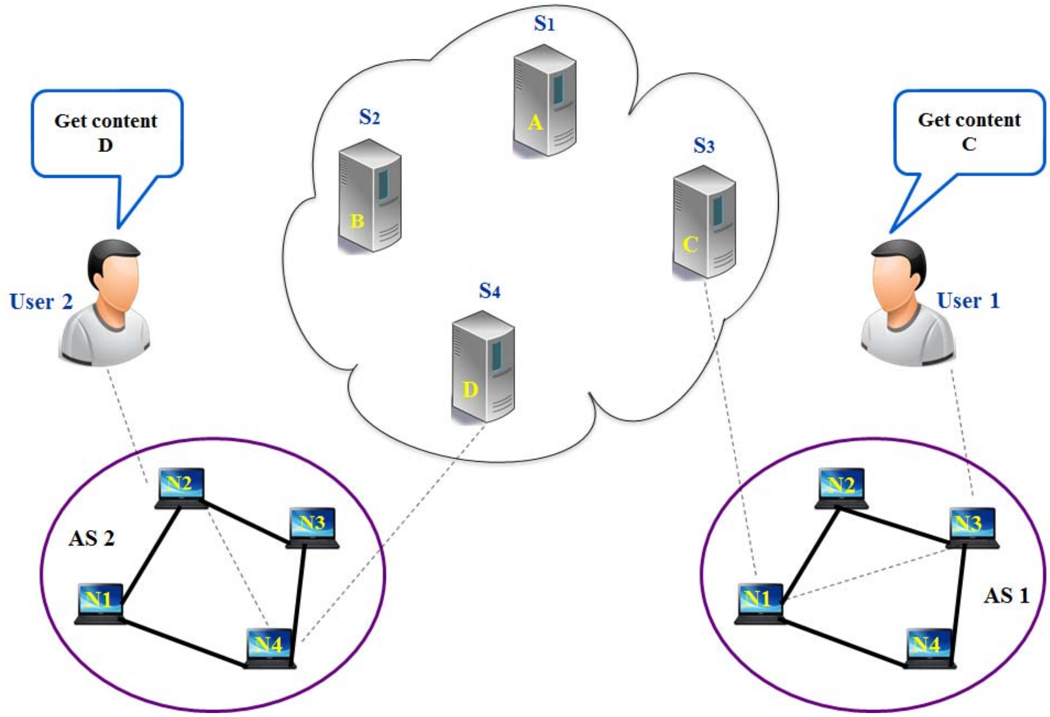


Fig. 2. ICN communication model: Adapted from [17].

original content. Another user (User2) in AS2 requests a different content, i.e., D, where its copy is available at node N4, thus, the request is satisfied through node N4 in AS2. The figure has four servers: S1, S2, S3, and S4. The network can assure the request with information from any node having a copy of the item, enabling application-independent and effective caching as a component of the system administration. This decoupling in time and space between publisher and subscriber needs that data objects carry security metadata for confirming the authenticity and integrity of the objects. The ICN research community accepts that named data, rather than its physical location, is the key component of routing in ICN. One of the leading proposals, CCN [38], has entirely changed information transmission by sending content requests and receiving through name at the network layer itself.

According to the ICN Research Group (ICNRG) at the Internet Research Task Force (IRTF) [6], data manipulation and distribution have been the business of the current Internet. Therefore, it is indispensable to diminish the challenges causing threats to the future Internet. Among the objectives of the ICNRG, the recent drafts [39], [40] include: The proposal and creation of scalable routing schemes, name resolution, cache management, and performance evaluation of the ICN paradigm. These along other challenges make ICN caching noticeable to conform with the recent drafts and recommendations of the ICNRG. Part of the motivation is that, when queries are served, a local caching node may have a copy in its content store (CS) so that if a new request arrives for that content, it can be satisfied locally rather than forwarding to the main server. This approach will increase content hit rate and reduce bandwidth utilization as well as content retrieval delay [41]. In this case, ICN has gained the popularity to be considered a vowing nominee for the future Internet architecture.

III. Related Work

In the literature, very good ICN survey papers exist, but because of their broad scope they treat caching strategies either partly or superficially. This paper covers ICN caching in more detail as compared to other ICN surveys (e.g., [42] and [43]). Besides, some other surveys are available, such as [44]–[46], however, as compared to those surveys, this article investigates the most recently designed caching strategies in line with their particular contributions and limitations. In addition, we have extensively simulated all caching strategies, as in [46]. The scope of related surveys is presented in Table I.

However, we believe that the performance of ICN cannot be examined only with cache hit, which is done in [46], because some mechanisms may increase the cache hit rate but affect the other metrics. Therefore, we have investigated the performance of caching strategies in terms of cache hit, stretch, and eviction ratios. These metrics were chosen for the performance evaluation because if a content is placed far from the users then the cache hit rate may be increased there but the stretch is reduced, which can cause content retrieval delay and thus the users' quality of experience is severely degraded. Also, some strategies evict the cached contents (even the most popular contents) very rapidly and thus the overall network throughput is reduced [47]–[49].

Moreover, we have also classified the strategies into different categories on the basis of their cache deployment and content caching, i.e., what particular content should be cached and what is the appropriate position of a caching

TABLE I
COMPARISON OF RELATED SURVEYS

Survey Paper	Reference	Contributions	Limitations
1	Zhang et al. [42]	Discusses the working of caching strategies in terms of content placement and eviction.	i. No performance evaluation. ii. Description of those strategies which are designed before 2013.
2	Ioannou and Weber [43]	Discusses ICN proposals and their respective architectures with a brief introduction of a few caching strategies.	i. No performance evaluation. ii. No deep explanation of caching approaches with respect to caching decisions and eviction operations.
3	Ioannou and Weber [44]	Discusses the on-path caching issues with the description of available policies for on-path caching.	i. No performance evaluation. ii. No description of ICN research challenges.
4	Abdullahi et al. [45]	Discusses the issues in four ICN architectures, namely DONA, CCN, NetInf, and PSIRP, with regard to cache placement using reactive, adaptive, proxy, and reverse-proxy models.	No explanation of caching strategies except a brief introduction of a few approaches.
5	Zhang et al. [46]	i. Discusses the ICN caching policies in depth, which are designed before 2015. ii. Evaluates the performance of those policies using simulations with respect to cache hit.	i. Caching in IoT-based ICN is mentioned as a challenging issue, whereas, some approaches have been designed for this kind of caching in 2015-17, which are not discussed here. ii. The strategies are evaluated only against cache hit ratio, while the performance of ICN caching strategies needs to be checked against other metrics, such as stretch and eviction operations, so that the best strategy may be selected for the ICN deployment.

node. Our classification is based on the content popularity, content caching location, the number of caching nodes, and the caching environment, i.e., ICN-based Internet and ICN-based Internet of Things (IoT).

Furthermore, the retrieval of multimedia applications is increasing exponentially [14], thus, we believe that testing the performance of ICN caching strategies by accessing multimedia applications is more appropriate. Therefore, we have used a Facebook topology [50]–[52] in the simulations for downloading multimedia contents. In addition to describing the aims and fundamental ideas of different caching strategies, it discusses the major unresolved research challenges in ICN caching which need further consideration by the research community.

IV. The General Idea of ICN

This section first describes the ICN communication model along with its major components and then illustrates the workflow and architecture of the common ICN approach.

A. Communication in ICN

The ICN architecture is different from that of the IP architecture. The existing Internet architecture follows a source-driven approach, i.e., a connection is established between the source and destination before any data transfer. The main objective of ICN is to find, transfer, and distribute contents as opposed to the reachability of end users and the establishment of conversation among them [53], [54]. In ICN, the customer requests content without knowing the providing host, and the route is established by the receiver to the customer, i.e., the communication follows a receiver-driven approach and the data follows the reverse route. The system is then responsible for mapping the requested data and its location. One of the applicable aspects of ICN for providing efficiency is naming, i.e., content must be named such that it should be independent of the node providing the content. However, the key point in ICN is accessing contents from the publisher(s), i.e., either the contents should be accessed by name resolution or name-based routing. The contents in ICN are recognized through such identifiers which are location independent and are not used for content forwarding.

The main features of ICN named contents are content level security and minimization of content retrieval delay. For this reason, two naming approaches have been proposed in the literature, i.e., flat names and hierarchical names [55]. Nevertheless, ICN naming may have some issues in the retrieval of multimedia and real time applications [24], [56]. Multimedia applications, which are requested and/or used by millions of users everyday, are video-on-demand (VoD), distance learning, interactive games, and so on, while the examples of real time applications are audio and video conferencing. These applications may raise the issues in content granularity [57], i.e., multimedia contents are made of several chunks which can be stored in different locations with different names [58], therefore, accessing these chunks brings up the issues of naming scalability [59]. The utilization of multimedia contents (both live and on-demand) is bound by time constraints for the content delivery [57]. Moreover, real time applications are more prone to delay and jitter, therefore, they should be avoided for attaining better user's quality of experience [59], [60].

The most popular and leading property of ICN is caching, where the router can store the content (for a time) passing through it (this storage depends on the cache size and replacement strategy) and provide them to the requesting clients. In ICN caching, the content is replicated through a caching strategy, therefore, the possibility of content delivery to the end user(s) is increased. Generally, the content in ICN is requested by user(s) without having information of the providing host. This is based on the receiver-driven approach, i.e., the path is established by the publisher to the subscriber and the data is forwarded on the reverse path. Mapping between the requested content and its location is then the responsibility of the network [53].

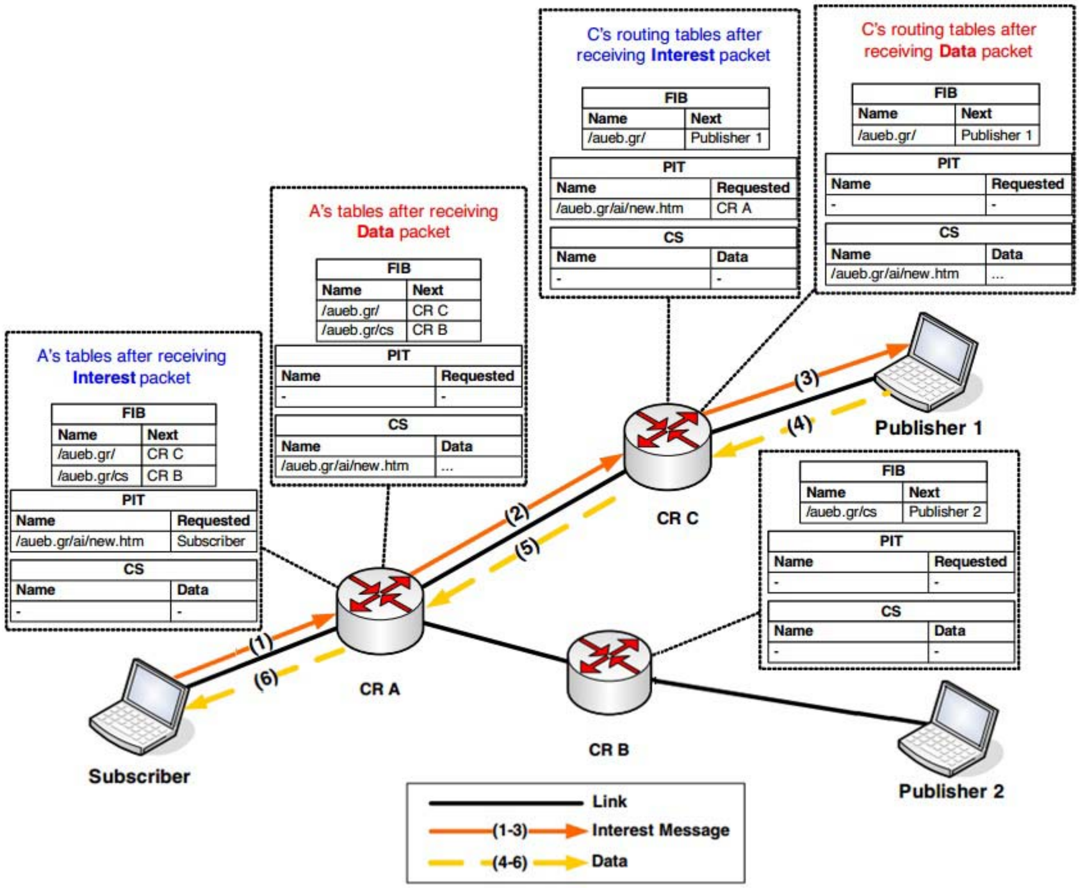


Fig. 3. CCN-based ICN architecture [1].

B. A CCN-Based ICN Architecture

CCN is one of the most dominant techniques for ICN. The workflow of CCN-based ICN approach [1] is shown in Figure 3. Two different types of packets are used for communication in CCN, i.e., INTEREST and DATA packets. To request content items, the subscribers send INTEREST packets, which are received in the form of DATA packets, with both kinds of packets carrying the name of the requested/responded content item. Content Routers (CRs) then forward all packets hop-by-hop maintaining three data structures: The Forwarding Information Base (FIB), the Content Store (CS), and the Pending Interest Table (PIT). The FIB forwards packets to the exact data source, the CS temporarily caches the received data packets, and the PIT tracks a group of interfaces from which the pending INTEREST packets have arrived. In simple words, it contains the names of INTEREST packets for which DATA packets are expected.

The subscriber requests with an INTEREST packet for the name /aueb.gr/ai/new.htm (arrows 1-3). If entry with the exact name is found in the PIT, the node discards this INTEREST and records the incoming interface to this PIT, successfully developing a multicast tree for the content item. This packet is sent back to the subscriber(s) in a hop-by-hop fashion, based on the status maintained in the PITs.

When a DATA packet is received, a router first caches the matching content item in its CS and then performs a long-prefix match on its PIT to trace an entry corresponding to the DATA packet. In case the PIT entry records more than one interface, the DATA packet is duplicated, and hence multicast delivery is achieved. Eventually, the router sends the DATA packet to these interfaces and discards the entry from the PIT (arrows 4-6). If no matching entry is found in the PIT, the router simply deletes the DATA packet as a duplicate.

V. Caching Terminology

Caching means to save data at some location(s) and then use next time if needed. In Web caching the content copies are saved near the subscriber so that they are accessed in no time when required [79], [80]. The most important advantages of caching are 1) to improve the availability of bandwidth by restricting the transmission of redundant contents, 2) to reduce content retrieval time, and 3) to save bandwidth cost. However, the main issue is to know that which Web contents are cacheable. To tackle this issue, the hypertext transfer protocol (HTTP) is enforced with some specific constraints to determine that which contents are worthy of caching. Caching may be designed to have the capability of detecting the failure of server and boosting the resources of proxy caching [79]. Although proxy caching is an effective step for increasing the Internet throughput, it has some limitations [80]. For example, caches are placed near the original server, therefore, the Internet throughput cannot be improved largely in the time of

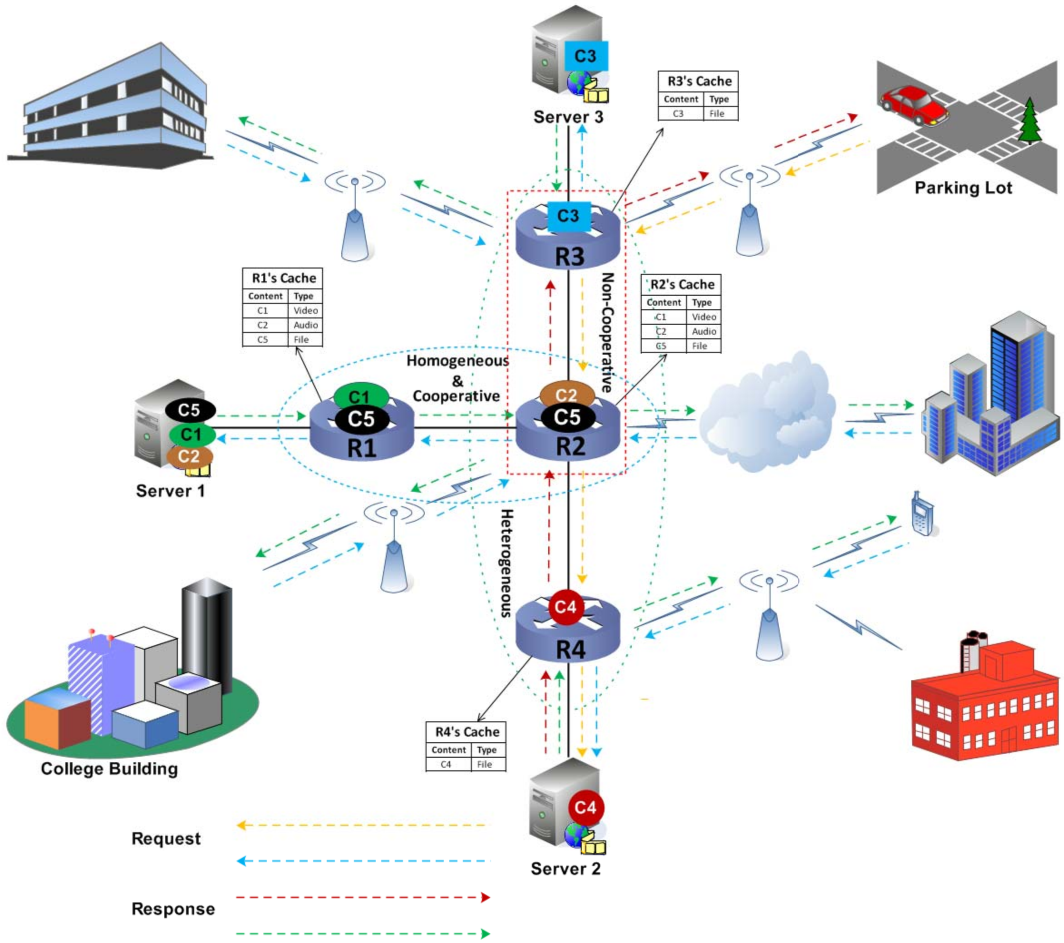


Fig. 4. ICN network with caching nodes.

network congestion which occurs due to exponential increase in the Internet traffic. In addition, the contents are cached on the basis of user demands, thus, the content providers may be forced for investing in high bandwidth connections and load balancing [79].

To overcome these limitations, the idea of Content Delivery Network (CDN) [81]–[83] was proposed in late 1990s [79], where computer systems are connected across the Internet to provide Web contents to the subscribers. Soon after introducing the idea of CDN, data providers started placing their Web contents on a popular CDN, named Akamai CDN [84]–[86]. However, as CDN is based on the IP address, despite several advantages, it has numerous shortcomings such as reliability, security, scalability, accessibility, performance, and affordability [79], [87], [88]. Therefore, for devastating the mentioned limitations, the ICN concept was introduced in 2009 [38], which is discussed in Section IV.

VI. ICN Cache Management

Caching is one of the essential modules of ICN [89], [90] which borrows its concept from the current Internet, presented in [81] and [91]–[97]. In-network caching is implicitly or explicitly studied in [98]–[104]. In this section, the importance of cache management and some of the designed approaches for in-network caching are explored.

To store contents precisely, that may change in time, some improvements are needed in traditional cache systems. One of the possible schemes to name these kinds of contents is to save their names, regardless of the fact that the contents change. If, for example, two routers store the content generated by two different sensors at different times, the information stored at different locations may be inconsistent [105]. Therefore, it is primarily important to deal with the outdated information. In this regard a time-stamp field could contribute, but it is not the proper solution as decades of cache consistency research revealed [82], [106]–[111].

Cache management in the time of congestion is a challenging issue in ICN. Reliable data delivery and flow control are usually viewed as considerably more of a challenge in content-centric publish/subscribe approach, where the delivery of data is analogous to multicasting [105]. ICN permits the network to provide content from any source where it is placed (including in-network cache), and satisfies the user requirements to send and receive data, regardless of its location. One of the most prevalent approaches of ICN is Named Data Networking (NDN) and is based on a query-response model [112].

TABLE II
CACHING CATEGORIES

Caching Strategy	Popularity-based Caching	Single Node-based Caching	Multi-level Caching	Location-based Caching	IoT-based Caching
Hash-routing [61]	×	×	×	✓	×
CIC [62]	×	×	×	✓	×
LCE [63]	×	×	✓	×	×
Prob [63]	×	×	✓	×	×
LCD [63]	×	✓	×	×	×
MCD [64]	×	✓	×	×	×
ProbCache [47]	×	×	×	✓	×
Breadcrumbs [65]	×	×	×	✓	×
CLS [66]	×	×	×	✓	×
OCPCP [67]	✓	×	×	×	×
CATT [68]	×	✓	×	×	×
Betweenness-Centrality [48]	×	✓	×	×	×
One-touch caching [69]	×	✓	×	×	×
NCCM [70]	✓	×	×	×	×
TLRU [71]	✓	×	×	×	×
WAVE [72]	✓	×	×	×	×
MPC [73]	✓	×	×	×	×
FGPC [74]	✓	×	×	×	×
Client-Cache [75]	×	×	×	×	✓
TCS [76]	×	×	×	×	✓
FMC [77]	×	×	×	×	✓
Sleep-based Caching [78]	×	×	×	×	✓

TABLE III
ANALYSIS OF CACHING IN ICN ARCHITECTURES

Architecture	Homogeneous Caching	Heterogeneous Caching	Cooperative Caching	Non-Cooperative Caching	Off-Path Caching	On-Path Caching
DONA [25]	×	✓	✓	×	×	✓
NDN [30]	✓	×	✓	×	×	✓
MobilityFirst [29]	×	✓	✓	×	×	✓
CONVERGENCE [35]	✓	×	✓	×	×	✓
COMET [37]	✓	×	×	✓	×	✓
SAIL [36]	×	✓	×	✓	×	✓
PURSUIT [34]	✓	×	×	✓	✓	×

Several approaches have been proposed for ICN cache management, such as cooperative, non-cooperative, homogeneous, heterogeneous, off-path, and on-path caching.

In cooperative caching, the network nodes share information about the content once they store it, while in non-cooperative caching, the information is not shared about the cached contents and therefore every network node makes a caching decision individually. In homogeneous caching, all network nodes on the publisher-subscriber path cache the contents once they pass through them. On the other hand, in heterogeneous caching, all network nodes available on the publisher-subscriber path do not cache the same contents.

Off-path approach is somehow similar to CDN server placement or conventional proxy caching, where the name resolution system (NRS) is informed upon the arrival of new contents [1]. On the other hand, in on-path caching, when a packet arrives, the router answers with a copy locally cached without contacting the NRS [103], [113]–[115]. To better understand this phenomenon, consider the example in Figure 4, where a user from college building requests the content (*C5*) which is available at Server 1. During downloading, both routers *R1* and *R2* cache content *C5* because they are lying on the publisher-subscriber path and therefore they are cooperative as well as homogeneous. Instead, router *R2* and *R4* are heterogeneous as another user from the parking lot requests a content available at Server 2, *R4* caches it but *R2* and *R3* (which are also available on the publisher-subscriber path) do not cache it. Routers *R2* and R3 are non-cooperative as they cache contents without sharing information about the cached contents. Thus, both of them may cache the same or different contents.

VII. EXISTING CACHE MANAGEMENT STRATEGIES

Content caching is a great concern of the ICN because the ICN cache strategy is still in initial stage and has a lot of space for the research community. Numerous cache management strategies have been developed, which are divided into different categories, as presented in Table II. This classification is done because *cache deployment* and *content caching* are the main issues in ICN. That is, which content should be cached and what is the appropriate position for content caching. For this purpose, several researchers have designed a number of caching strategies, which show either the importance of a particular content for caching or the best possible positions of caches. These strategies are explained in the following subsections. In addition, Table III presents the generic view of the ICN architectures that which of the caching categories can be easily achieved in a specific architecture. Tick marks or cross symbols against the caching categories in Table III do

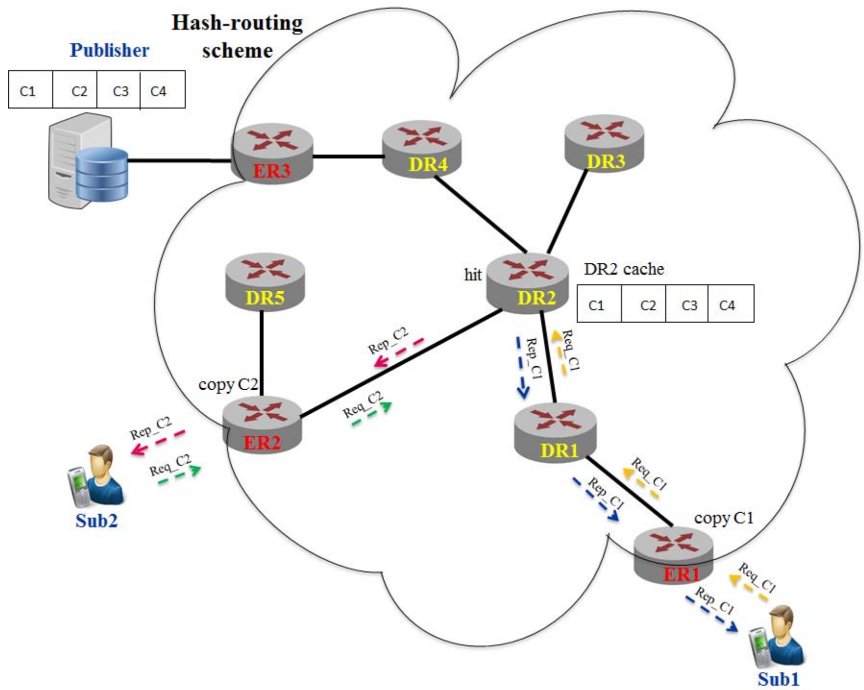


Fig. 5. Hash domain routing architecture.

not mean that those caching properties are not supported in these architectures, but they require some additional information, for example, on-path caching in PURSUIT or off-path caching in the other architectures.

A. Location-Based Caching

In location-based caching, the contents are cached at some specified node(s) on the basis of a particular function. In this kind of caching, several strategies have been designed, which are discussed in the following subsections.

1) Hash-Routing: Hash-routing [61] schemes are proposed for off-path caching. According to hash-routing schemes, if a content request arrives at an edge router, it calculates a hash function that maps the content identifier to a certain caching router and forwards the request to that particular router. The router then returns the requested content to the user if it is maintained in its cache, otherwise forwards it to the original source.

Similarly, when a router forwards the requested content to the user, only that router can cache it which is associated to the content identifier. The main objective of the hash-routing schemes is to remove caching redundancy as the hash function indicates to cache the content where it is found.

In Figure 5, the subscriber (*Sub1*) requests a content, which is cached at the domain router (i.e., DR2), where the request enters the domain through the edge router 1 (ER1). Router DR2 returns the content to *Sub1* and a copy of that content is also cached at the edge router ER1 because the request was forwarded through that router.

If the content item (entering the domain through the edge router ER1) is forwarded to router ER2 and the specified cache is at router DR2, then multicast delivery becomes more effective. Likewise, a symmetric delivery is preferred if the specified cache is at router DR5. In this case both approaches, i.e., multicast hash-routing and symmetric, can achieve higher cache hits. However, in both of these schemes the possibility of intra-domain link load will be potentially high.

2) Cooperative In-Network Caching (CIC): This is another off-path caching strategy. In the cooperative In-network Caching (CIC) [62], a content is divided into different chunks and is cached at more than one node such as router. For instance, a content is 12 chunks: each chunk is of the same size, and the cache capacity of a router is only to store six chunks, then this content is divided into two same size chunks and cached at two different routers. To better understand the concept of CIC, consider Figure 6, where each route has a cache capacity of 7 chunks, but the chunks are cached based on *modulo 3* division. In other words, a router does not cache all chunks of a content in a sequence, but a part of those chunks are stored according to *modulo k*. Let us say that $k = 3$, then router C will cache chunks 0,3,6,9,12,15,18 and router *B* and A will cache chunks 1,4,7,10,13,16,19 and 2,5,8,11,14,17,20, respectively.

Now, if the subscriber requests a content whose part is chunk 15 – cached at router *C*, it will not be forwarded to the server, but router *C* will satisfy it locally. Similarly, if a next request arrives for another content whose part is chunk 5 – cached at router *A*, router *C* cannot satisfy it, therefore, it will be forwarded to router *A*. Hence, both the requests are satisfied by the local machines because of this cooperation. The CIC is an off-path caching strategy because the contents are cached at the routers which are not available on the publisher-subscriber path.

The caching diversity which is the amount of different chunks in the cache can be improved by CIC with avoiding

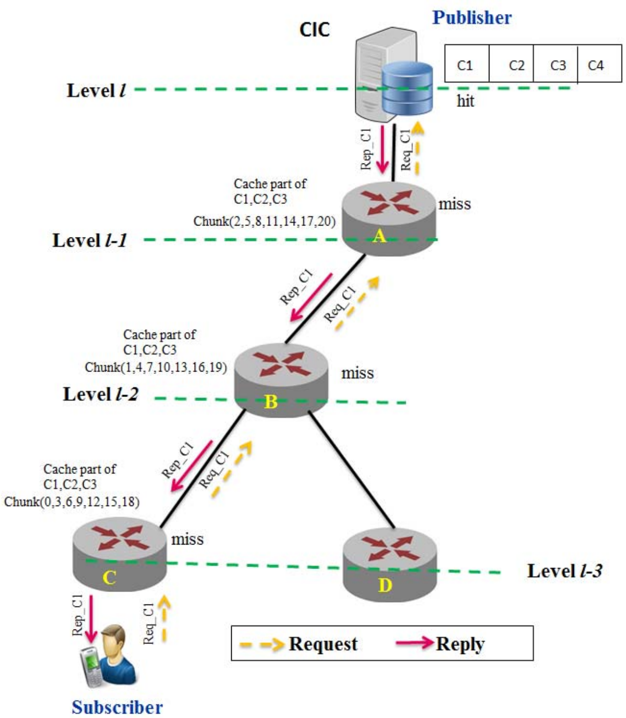


Fig. 6. CIC Topology.

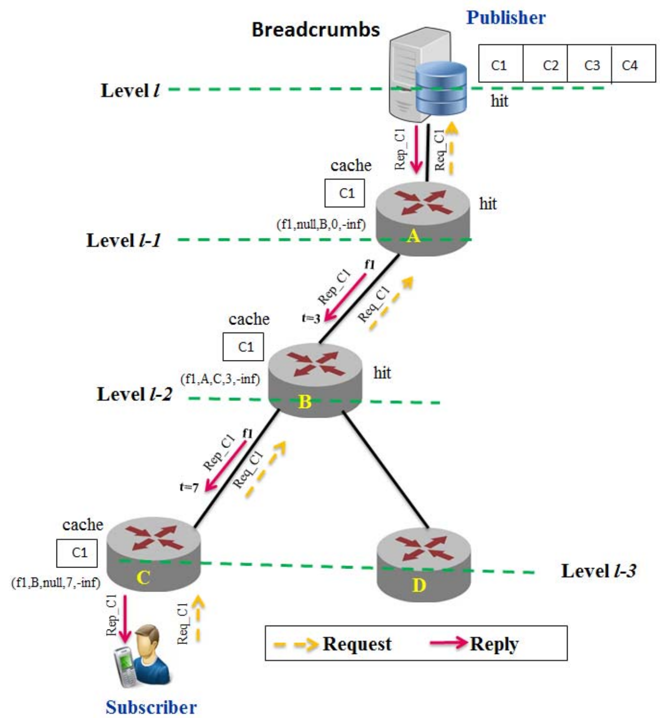


Fig. 8. Breadcrumbs topology.

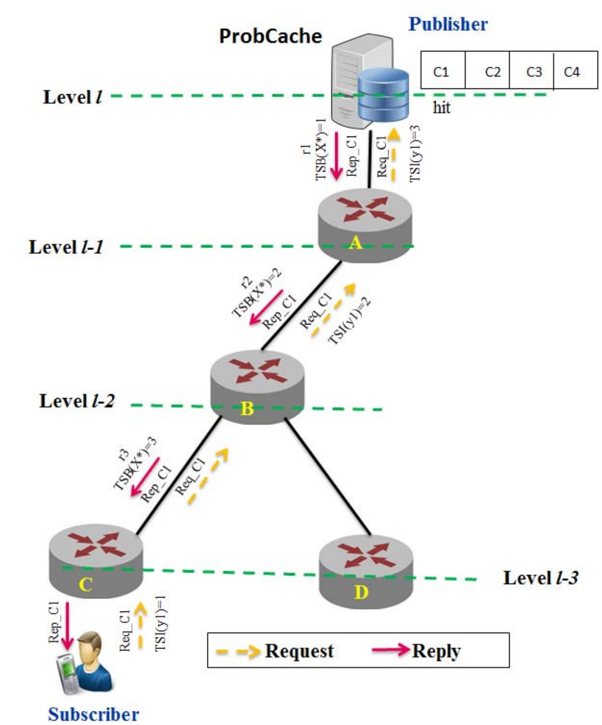


Fig. 7. Design topology of ProbCache.

caching redundant chunks. Moreover, the average response time: The turnaround time between the request initiation and the receiving of the requested chunk, can also be enhanced. However, the CIC adds two extra tables, named Collaborative Router Table (CRT) and Collaborative Content Store (CCS). CRT is used to keep track records of all cached chunks along with their IDs, whereas CCS is used for keeping the sequence numbers and names of all the cached chunks at every router. These operations involve some extra computational overhead and therefore increase the searching time in case some chunks need to be evicted for caching a new arrived content.

3) Probabilistic Caching (ProbCache): In the ProbCache [47], as shown in Figure 7, the *Time Since Inception* (TSI) is a field required in the ICN *request message* headers, while the *Time Since Birth* (TSB) is an important field in the *content message* headers. Where *r* represents router, *n* is the number of caches on the path, and *X* is the hop-distance from the server. In the given topology, a user is connected to the server at a distance of four hops. When a cache hit occurs, the TSB value of the content packet is set to 1, while the TSI value is replaced by that of the *Request* packet.

This policy is considered as a *resource management* approach to in-network caching. The ProbCache attempts to maximize the amount of different content items stored along a communication route. In other words, it tries to reduce redundancy between network caches. Thus, the probability of finding content along the path becomes higher for subsequent requests. But, the *Hop decrement* (the percentage of *content request hops* and the *path hops* between the user and server), which is an important parameter for expediting the process of content access, will become lower in the ProbCache. The lower the *Hop decrement* the higher the content delay, and vice versa. The *Hop decrement* in ProbCache decreases because it frequently replaces contents at nodes near the user(s).

4) Breadcrumbs: The Breadcrumbs [65] is a transparent and best-effort simple searching and content caching technique. Each breadcrumb is a *5 − tuple* entry, indexed by a

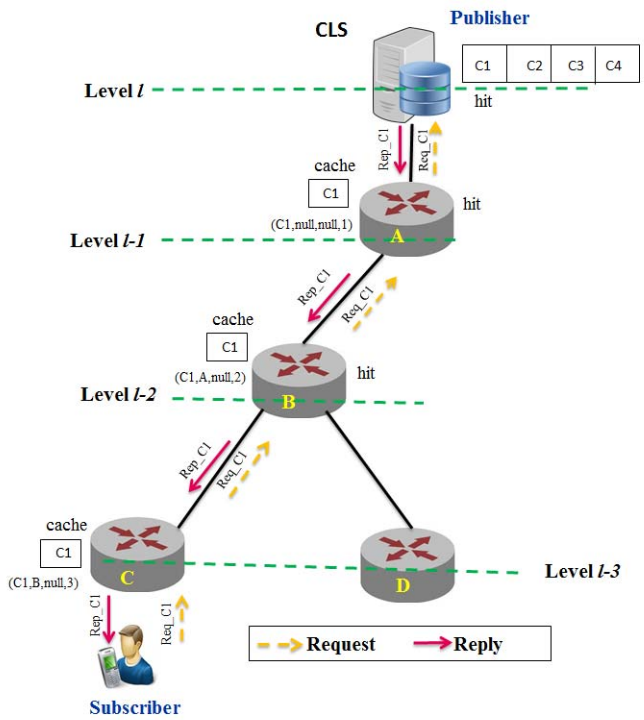


Fig. 9. Operation of the CLS scheme.

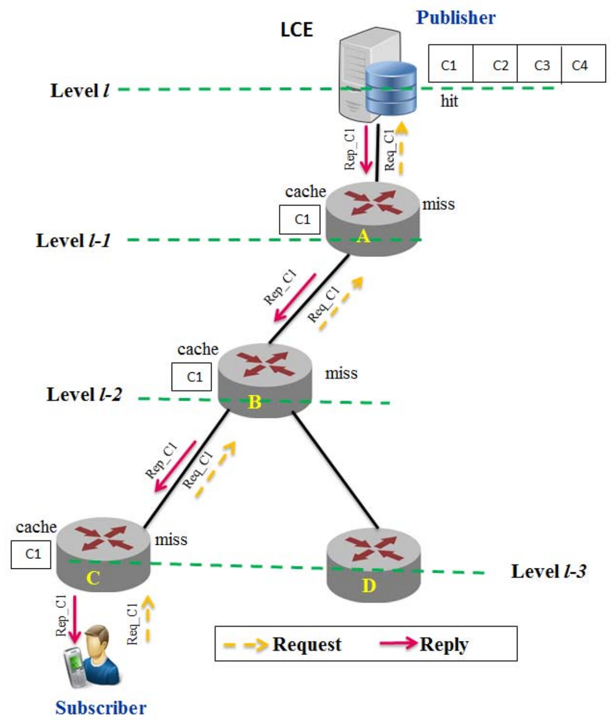


Fig. 10. Operation of LCE.

global content ID. This ID holds a set of information, i.e., 1) the ID of node to which the content was pushed down, 2) the ID of node from which the content was received, 3) the most recent time of the requested content, and 4) the most recent time of the forwarded content (see Figure 8).

As the content is downloaded, a trail is created and maintained at each router for caching the routing history. This trail will route the subsequent request downstream for the same content towards the router's directory, instead of upstream towards the server. Similarly, to check whether searching downstream or not, a time threshold is used in this scheme. One of the main advantages of a downstream request routing technique is to minimize the server overhead. Another benefit of breadcrumbs is to reduce the content download time if it is found downstream.

However, if the content is evicted at each router, the request may suffer missing downstreams, and hence the overall content delay, during downloading, will undoubtedly increase.

5) Chunk Caching Location and Searching Scheme (CLS): The main objective of CLS [66] is that a hit at level l, as in the MCD (discussed later), drags the requested chunk down to the *l-1* level node (see Figure 9). When the content is downloaded, as in the Breadcrumbs, every router creates a trail for caching the routing history. Each chunk is a $4 - tuple$ entry, indexed by a global chunk ID. This chunk ID holds a set of information, i.e., 1) the ID of a node from which the chunk was received, 2) the ID of a node to which the chunk was pushed down, and 3) the number of hops between this router and server.

Each router adapts the PIT to cache the trail information of formerly stored chunks. The major difference between the trails of Breadcrumbs and CLS is that the Breadcrumbs trail is generated at the time the content is pushed down, while in the CLS the trail is maintained at the time of chunk caching. This guarantees that the CLS saves cache space as it deletes the trail on time and consequently finds the chunk if the search is downstream.

However, according to the CLS trail, an intermediate node cannot judge whether the server is close or the cached copy. Therefore, it cannot ensure the placement of chunk at the nearest node to the user.

B. Multi-Level Caching

In multi-level caching, the contents are cached at more than one location in the network. In this category, the most prominent caching strategies are Leave Copy Everywhere (LCE) and Prob, which are discussed below.

1) Leave Copy Everywhere (LCE): LCE is considered as a multi-level caching technique. In Figure 10, the user (subscriber) requests a content which is available on the server (publisher). The request is forwarded through router C, B, A to reach the publisher. In the LCE [38], [63], when a hit occurs either at the publisher or a *level l* cache, a replica of the requested content is stored in the caches of all nodes between the publisher and subscriber(s).

The LCE caches contents at all on-path nodes, therefore, if a new request arrives for the one-time-accessed content, it is not forwarded to the publisher but the in-network node replies with the requested object if that has already cached it. Through caching a content at each in-network node the bandwidth utilization can be improved effectively because the requests are not forwarded to the publisher but satisfied locally. However, due to redundant replicas of the same content, the

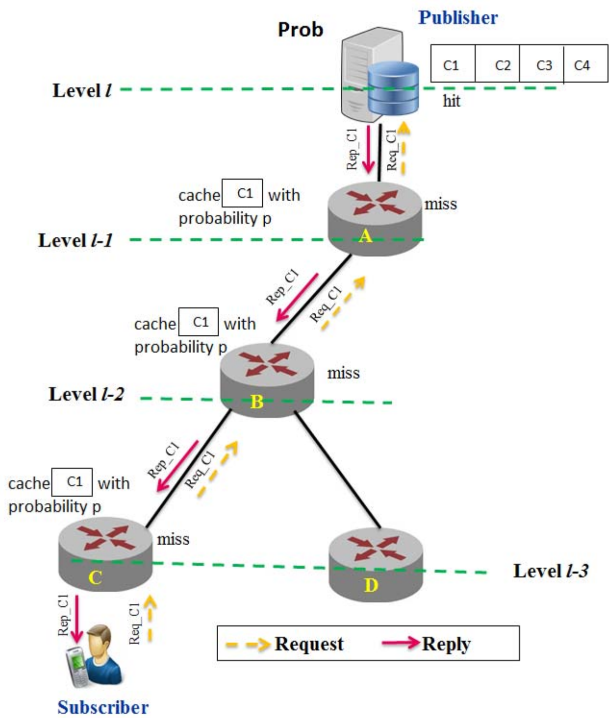


Fig. 11. Operation of Prob.

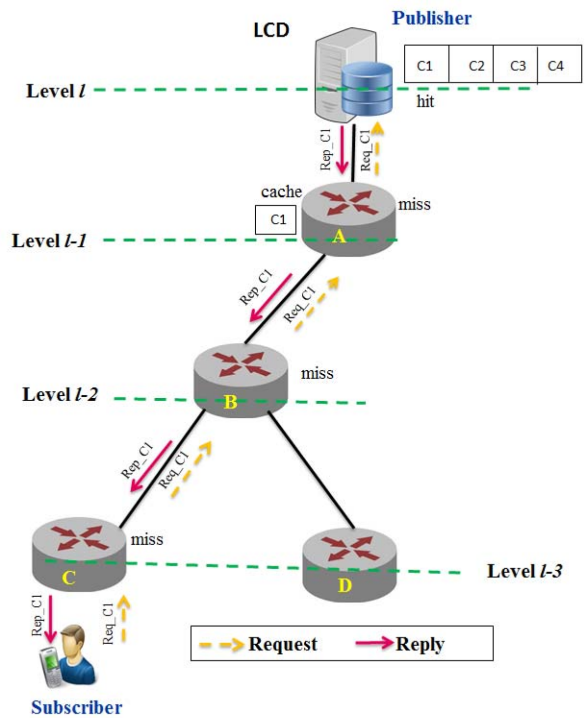


Fig. 12. Operation of LCD.

cache of all in-network nodes overflow, and thus new arrived contents will never participate in the cache hit.

It seems that caching content at all nodes will increase cache hit ratio, but this phenomenon is impractical because when the cache of a node overwhelms and a new content arrives then it needs to replace one of the cached contents. Most of the strategies follow Least Recently Used (LRU) or Least Frequently Used (LFU) policy for content replacement [116], therefore, the evicted content may be a popular one, which is possible in both LFU and LRU replacement. Now, in case of a new request arrival for that evicted content, it will not find the content locally, therefore, the request will be forwarded to the server. Consequently, it will affect the overall network performance with respect to cache hit rate, content retrieval delay, and bandwidth consumption.

2) Prob: Prob [63] is a randomized form of the LCE strategy, where each node can cache a copy of the requested content on the path from the server or node where hit occurred to the requesting host (see Figure 11). A node in the middle of the network does not cache a copy with probability *1-p*, but can keep only a local copy with probability *p*. The operation of Prob(*1*) is similar to that of the LCE strategy.

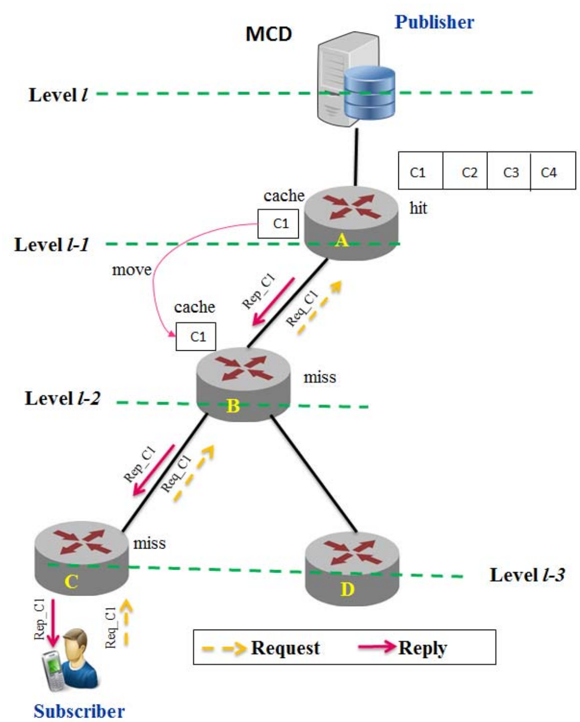


Fig. 13. Operation of MCD.

C. Single Node-Based Caching

In single node-based caching, a content is cached at only one location in the network. In this kind of caching if a new content arrives and the memory is full, one of the cached contents is replaced (through a particular method) to accommodate the new arrived content. several strategies lie in this category, which are explained below.

1) Leave Copy Down (LCD): In the LCD [63], a copy of the requested content is stored merely at the $(l - 1)$ level node, i.e., the one which is located instantly down the node where hit occurred, as shown in Figure 12. To forward a content to the leaf node, unlike LCE, the LCD requires multiple requests, with each request forwarding a fresh copy of the content one hop closer to the client.

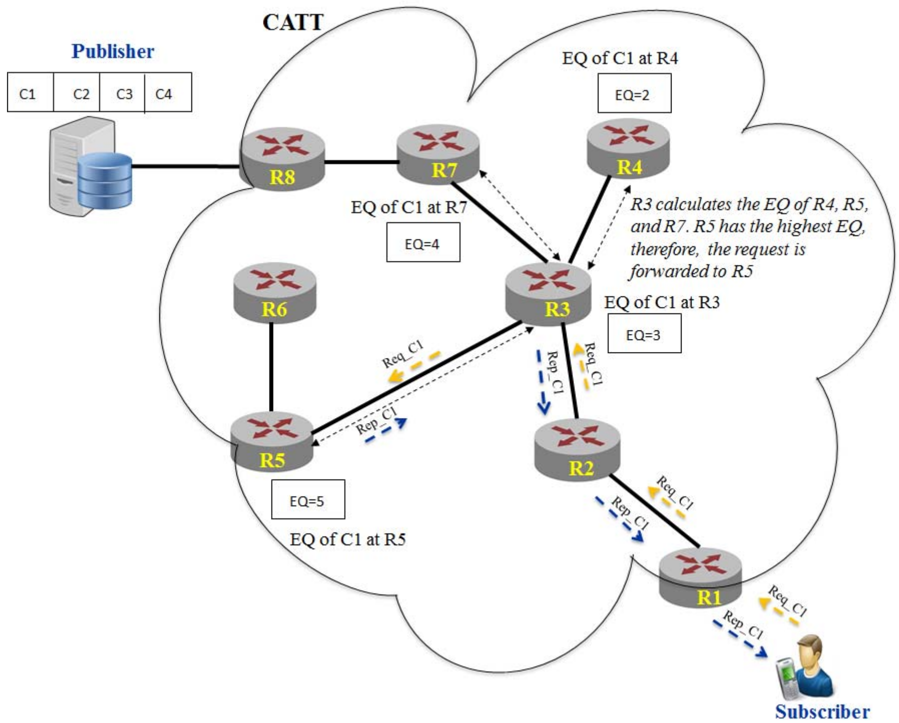


Fig. 14. A CATT topology.

With caching contents only at one node, i.e., ($l - 1$) level node, the LCD increases the cache hit ratio and effectively utilizes the cache space. However, if some contents are requested frequently (popular contents), then at some time all nodes on the link will cache these contents and hence, cache of the nodes will become overwhelmed and the space will not be available for the new arrived contents. The LCD also uses LRU for content replacement in case the cache becomes full, therefore, excessive eviction operations happen and the new arriving contents replace the popular ones. Consequently, all new requests for those evicted contents are forwarded to the server and in turn a maximum utilization of bandwidth is experienced [41].

2) Move Copy Down (MCD): The functionality of MCD [64] is analogous to that of the LCD, but the only difference is that a hit at level l shifts the requested content to the underlying cache. In Figure 13, as hit occurs at level *l-1,* the content is moved to level *l-2*. In this approach, no deletion is required when a content is hit at the main server, but the deletion takes place only if the requested content is hit at a location rather than the original server. The main objective of the MCD is to minimize the content redundancy between the server and the requesting host.

Looking at Figure 13, the subscriber requests a content which had already been accessed by some user(s) and cached at router A. As the request reaches router A, the hit occurs there and router A replies with that content. Here, unlike LCD, the copy of that content is moved to the down node, which is router B, and the cached copy at router A is deleted. Furthermore, if the same content is requested again by this subscriber, its copy at router B is evicted and moved down to router C. This process efficiently utilizes memory as well as decreases the stretch. However, if another subscriber - connected to router D, requests the same content, the request reaches router B via router D, where no copy of that content is available. Thus, the request is sent to the publisher and consequently the average cache hit rate is reduced and content retrieval delay is considerably increased.

3) Cache Aware Target Identification (CATT): In the CATT scheme [68], [117], the content is cached at a single node on the publisher-subscriber path. The CATT introduces the expected quality of contents to help a router forward content requests. If a content is cached at a router, its expected quality is supposed to be E_q, while the expected quality for a non-cached content, denoted by $E(NC)$ is calculated as:

$$E(NC) = \frac{E_q}{R_h + 1} \tag{1}$$

where R_h is the number of hops between the caching router and the current one.

If a router caches a content, it multicasts the content related information to its neighbors. As a neighbor router receives this information, it first calculates the previous expected quality of the content and then sets its new expected quality.

In this scheme, on the edges of an AS some CATT nodes (CATNs) are deployed as caching nodes. CATNs have three major modules, namely cache, repository, and routing. The provision of caching capabilities for all CATNs is responsibility of the first module, i.e., cache. A caching decision is made when a content is downloaded from the publisher to the subscriber. The second module, repository, manages the cache where a subscriber publishes contents. The last module,

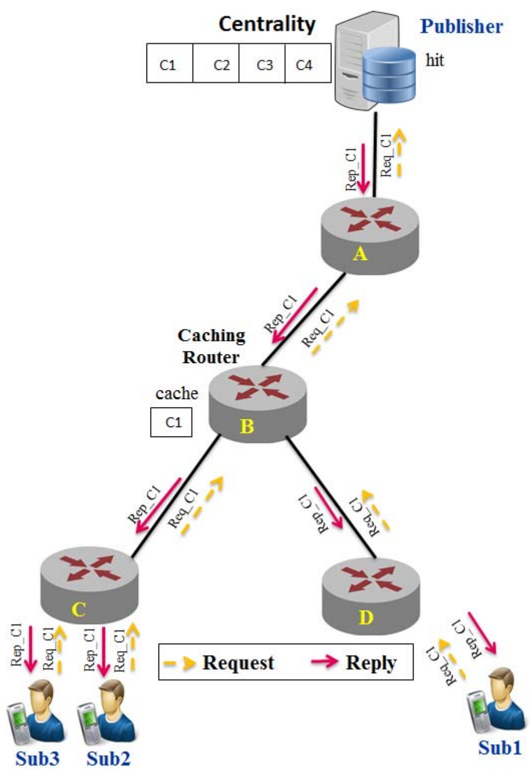


Fig. 15. Caching in Betweenness-Centrality.

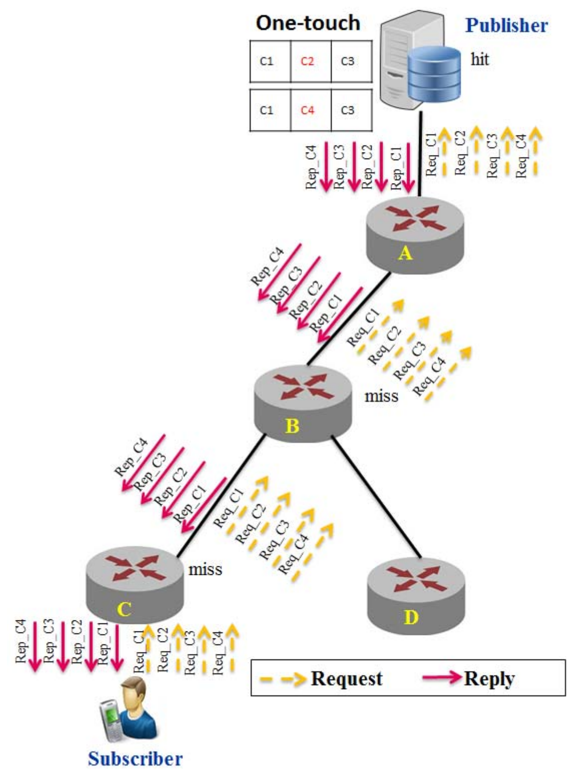


Fig. 16. One-touch caching scheme.

routing, forwards the content requests based on their names. In the CATT scheme, instead of FIB, the Potential Based Routing (PBR) is used for content retrieval. If a node receives a content request and it is not stored in its cache, it compares the expected quality (EQ) of that content with its neighbor's PBR and forwards the request to that neighbor which has the highest EQ for this content.

In Figure 14, router *R3* receives a content request from the Subscriber via router *R2* and finds that its highest EQ is at router *R5*. Therefore, it forwards the request to router *R5* rather than router *R4* or *R7*.

As the CATT caches contents at the edges of AS, it reduces content redundancy and retrieval delay [46]. However, if the cache of edge routers overflow then the new contents will replace the stored ones, hence, the increase in cache hit ratio cannot be guaranteed.

4) Betweenness-Centrality: In the Betweenness-Centrality [48], the content items are stored only once along the path, particularly at the router having the highest betweenness centrality, i.e., the router having the maximum number of shortest routes traversing it. Consider the example in Figure 15, a subscriber (*Sub1*) requests a content which is available on the publisher. The publisher will reply with the requested content and a copy of it will be cached at router *B* which has the highest betweenness centrality. Furthermore, if other subscribers (*Sub2* and *Sub3*) request the same content, router *B* will satisfy their requests locally rather than forwarding to the publisher. In case more than one router has the highest betweenness centrality, then the content is cached in a router located near the customer.

In this strategy, caching decisions are made on the basis of betweenness centrality, but it is challenging for a router to find the value of its betweenness centrality if it is deployed on every mobile ad-hoc network. In addition, if the cache of router *B* becomes full and a new content needs to be cached, the LRU policy will replace one of the cached contents, which may also be popular, and the subsequent request(s) for that content will be forwarded to the publisher. This operation can reduce the cache hit rate as well as maximize the bandwidth utilization.

5) One-Touch Caching: In the One-touch caching scheme [69], the most recently requested content is preserved in the server and existing content is randomly evicted if the cache space is full. If free cache space is available in the server, all the requested contents are accumulated there. The requested content is replied from the cache server if it is maintained in its cache.

The content management approach for one-touch caching scheme is shown in Figure 16. In the given example, the Publisher's cache can accommodate only three contents. Until the last time slot, which is 3^{rd} in the given figure, no deletion took place as the Publisher had storage capacity. When a content is requested, it is replied from the Publisher if available, as presented in the first three time slots. However, if a new request arrives after the 3^{rd} time slot, one of the stored contents needs to be evicted from the Publisher's cache. This content eviction takes place in a random fashion. It is shown in the figure that content $C4$ is cached and the stored content, i.e., $C2$, is evicted. This eviction of content $C2$ is in a random way.

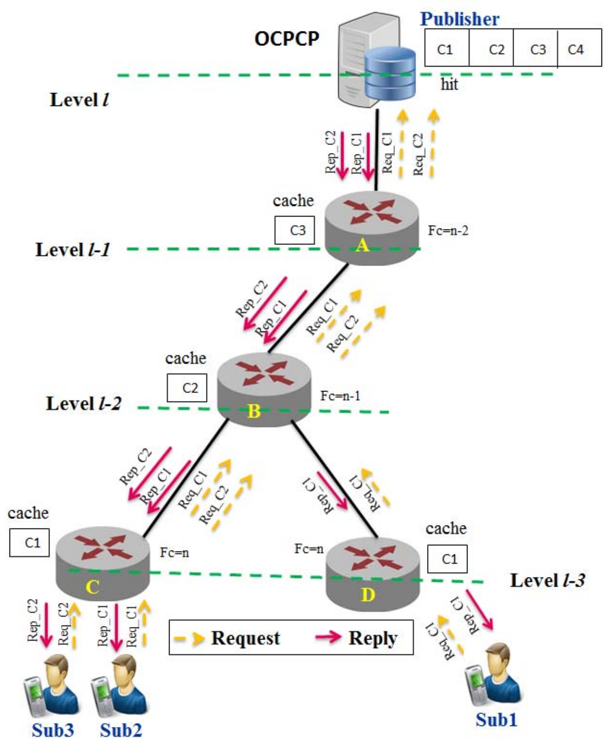


Fig. 17. An OCPCP topology.

Due to the simplicity of One-touch caching scheme, there is no need for sorting operation, tedious caching, and replacement processing. One-touch caching has little overhead as compared to LRU and LFU, but as the contents are not cached locally in this scheme, all requests are served by content servers and thus maximum bandwidth may be utilized during request to and/or reply from the server(s). As a result, unavoidable content delay may occur.

D. Popularity-Based Caching

In popularity-based caching, the contents are cached on the basis of their access time, i.e., when the predefined threshold value of a content reaches then it is recommended for caching at some network nodes [118]. In the following subsections, some of the available popularity-based caching strategies are discussed in detail.

1) Optimal Cache Placement Based on Content Popularity (OCPCP): In the OCPCP strategy [67], the incoming content's popularity is computed based on the stored contents and the new content is stored on the basis of its popularity value. The OCPCP takes a caching decision by just considering the records of content requests at a single node. That is, the more frequency of requests for a content, the higher probability of next requests. In other words, if the amount of content requests is high, it has a higher popularity and is therefore considered for caching.

Consider Figure 17, initially the cache of all routers is empty. The subscriber requests a content which is available on the publisher. The request reaches the publisher via routers *C,B,A*. As the popularity of a content (which is the number of requests) increases, it is cached on the router near the subscriber, i.e., router C and D in the given figure. When the network gains stability, router C and D cache the most popular content (*C1*), denoted by $F_c = n$, where F_c is the frequency of content requests. Routers B and A would cache the second and third highest popular content, respectively. The OCPCP counts this frequency of content requests through a data structure called Request Record Table (RRT). This calculation is based on the following equation:

$$F_c = \frac{\sum_{i=1}^{n} \delta_c(i,j)}{N} \tag{2}$$

where F_c is the total frequency of contents; N is the total number of requests; $\delta_c(i,j)$ is a function if the new content c_i and the j^{th} content of RRT are equal then its value becomes 1.

Content placement in the OCPCP is based on the following Content Placement Algorithm (CPA):

Step 1: Store the incoming content if the cache has capacity, otherwise, calculate the popularity of F_{ci}.

Step 2: Check whether F_{ci} has the highest popularity, i.e., $F_{ci} = n$, if yes, follow Step 3, otherwise go to Step 4.

Step 3: Cache content F_{ci} on LRU basis.

Step 4: Update the value of content c_i in RRT.

The OCPCP caches contents only at a router located near the subscribers, hence, the content redundancy and the retrieval delay are minimized. This placement of contents near the subscriber(s) can also improve the network throughput in terms of bandwidth utilization as the requests can be satisfied locally instead of forwarding to the publisher. However, because of the rapid increase in the number of Internet users and demand for multimedia applications, the chances of filling router's cache are very high. As a result, the new arriving contents may replace the cached popular ones and therefore they will no longer participate in the cache hit ratio.

2) Time Aware Least Recently Used (TLRU): LRU is a widely deployed cache management policy. It maintains contents in cache according to their access time. If there is no free cache space and a new content arrives, it deletes the least recently used content to make space for the new arrived one. On the other hand, LFU policy sorts the queue in terms of access frequency. If the cache space is full and a new content arrives, it replaces the lowest frequency content of the queue.

However, due to the nature of LRU and LFU, there is a possibility of evicting the most popular content by a least popular one. To overcome this problem, the Time Aware Least Recent Used (TLRU) policy was proposed in [71]. It is a popularity-based content life time aware eviction policy, which is an extension of the simple LRU. In this policy, the time stamp of an arriving content is calculated locally by a caching node. The arriving content is cached if the average request time is smaller than the time stamps of the stored contents. The TLRU stores the content if space is available in cache otherwise it applies simple LRU on the cached contents to create space for the new arriving content. Unlike LFU and LRU which are non-protective eviction policies, TLRU is proposed to evict relatively less popular contents.

In this policy, as shown in Figure 18, the eviction order of the cache state can be changed at any instant of time without

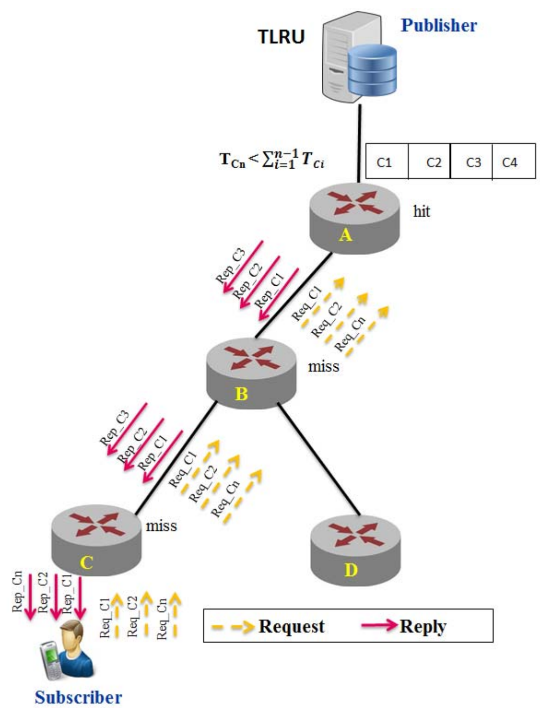


Fig. 18. TLRU eviction policy.

changing stored contents in the cache. In the given figure, a subscriber requests content C_i, where $i = 1, 2, \ldots, n$. The time stamps of these contents are calculated locally by the caching routers. After some time, if the time stamps of a new content (i.e., content n) is less than the average time stamps of all cached contents, such as $T_{Cn} < \sum_{i=1}^{n-1} C_i$, where T_{Cn} is the time stamp of content n and $\sum_{i=1}^{n-1} C_i$ is the average time stamps of contents stored in the local router, then content C_n is cached in the memory of router A.

The authors claim that TLRU outperforms LRU and LFU if space is available in cache otherwise the simple LRU policy is used to create space for the new arriving content. But according to Cisco Visual Networking Index [14], as the delivery of multimedia applications is becoming extremely popular on the Internet, the aggregation of all forms of video will be roughly 86 percent of the worldwide user traffic. Due to this exponential increase in the number of Internet users and demand for multimedia applications, there is a vivid chance of filling cache of CRs in a network very rapidly. In that case the TLRU will be no longer effective for content eviction.

3) Network Coding Based Cache Management (NCCM): The NCCM scheme [70] is developed to achieve near-optimal caching and routing solutions for ICN. Here, a novel framework based on the Software Defined Network (SDN) is proposed for cache management in ICN which jointly considers content routing and caching strategy through Linear Network Coding (LNC). Using this scheme, the problem of minimizing the network bandwidth cost is formulated through LNC. The linear combination of the original data chunks can be cached at multiple routers.

In order to obtain the best possible solution, content request statistics should be reported periodically with a fixed time period by all CRs. For popular content routing, the controller needs to discover the possible path and therefore configure flow tables in CRs on the path. When a request arrives at CR from its local end user, it will first check if the same content with the same sequence number is already requested and the data chunk is not returned, then it will not send this request. If not, the request will be forwarded through an outgoing interface when the number of data chunks that can be received from this interface is greater than the number of data chunks received from this outgoing interface. A CR will create a new coded data chunk (if it receives a cache hit) by randomly combining the cached data chunks and return it. Similarly, the CR will obtain the requested data chunks from one of the servers if it misses a cache. As the CR obtains the coded data chunks, this caches it and forwards to the requested user.

In Figure 19, the topology consists of 8 routers: $R1$ through $R8$, two subscribers: $Sub1$ and $Sub2$, one Publisher, and one controller. A user ($Sub1$) requests content $C1$ from the publisher via $R1, R2, R3, R7, R8$. The Publisher replies with the requested content on the reverse path. Another user ($Sub2$) requests the same content and the request is routed via $R5, R3, R7, R8$. As both requests are routed through $R3$, the controller instructs $R3$ (through $R4$) to cache content $C1$.

In the NCCM, cache management for ICN using LNC has been considered to improve the network throughput (i.e., to reduce the cost of network bandwidth by jointly considering content routing and caching strategy). However, the support for network coding may be an issue in implementation.

4) WAVE: In the WAVE [72], chunks are adjusted in cache based on the popularity value of the contents. The basic idea of WAVE is that, as illustrated in Figure 20, the chunks are organized based on the access count of the contents. Upon increasing the access count, the number of chunks to be stored is increased exponentially and disseminated more widely. In addition, WAVE does not require any prior knowledge of access patterns, therefore, it can be operated with any content routing strategy (as it follows merely the information from where it receives the content request). As no central server is used in this scheme, caching decisions are made independently at each individual router. Moreover, the number of content items is recommended to a downstream router by its upstream one, which is increased exponentially as the number of requests increases. If the downstream router is unable to cache the content by any reason, it simply ignores the suggestion and recommends the content to other routers on the same path.

In Figure 20, a content (having 7 chunks) is requested by a user (*Sub1*), which is available on the Publisher. The Publisher forwards the actual content to *Sub1* and marks the first chunk of the content ($Ch1$) so that it shall be cached at the level $l-1$ router (i.e., router A). The remaining chunks are forwarded to the requesting host (*Sub1*) without caching at any router. Now, if a new request for the same content arrives from *Sub2*, it is forwarded to the Publisher, however, the first chunk is served from router A, which has a copy of that chunk. Meanwhile, the Publisher will mark the second and third chunks ($Ch2, Ch3$) to be cached at router A, while router A marks the first chunk ($Ch1$) for caching at router B. Similarly, if other subscribers

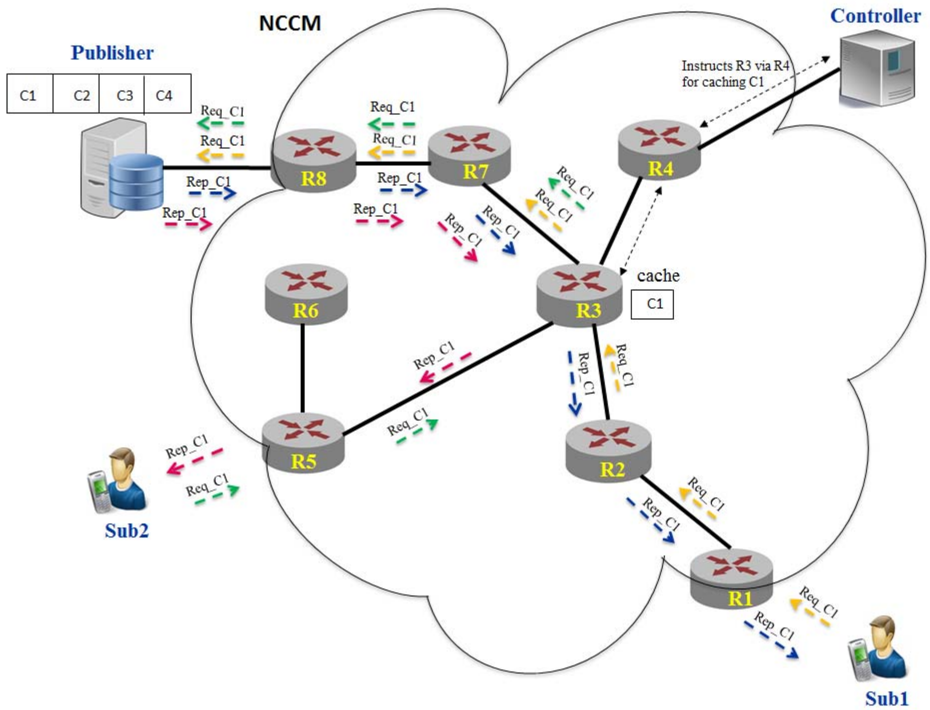


Fig. 19. An example topology of NCCM.

request the same content, the Publisher marks the next chunks for caching at router *A*, whereas route *A* suggests caching the next chunks at the underlying router, i.e., router *B*. Thus, a time will come that routers *A*, *B*, and *C* will have all 7 chunks of the requested content.

In the WAVE, an upstream router suggests what content chunks to be stored in its next downstream router. Consequently, as content is becoming popular, replicas of the corresponding chunks are becoming closer to the end users [113]. Although WAVE is an efficient caching scheme, it cannot eliminate the caching redundancy and is difficult to achieve good caching performance locally [119]. In other words, as distance grows, the performance is dramatically decreased [120] because content servers are involved in the recommendation [121]. Furthermore, for the adjustment of the number of contents to be cached, WAVE takes benefit of the popularity property of the content at each site in ICN and disseminates the contents in a hop-by-hop fashion. While with the exponential growth in mobile networks, the introduction of mobile ICN is of great interest in terms of content properties, for example, popularity, geographical and social characteristics. But these features (i.e., content characteristics and wireless access of mobile users) are ignored in the WAVE [122].

5) Most Popular Content (MPC): The MPC [73] is developed for the CCN caching, where network nodes store only popular contents. In this strategy, when subscribers make request messages for particular contents, every network node computes those requests and stores their values in a *Popularity Table* (PT). If the popularity threshold is achieved locally then the content name is labeled as popular. Moreover, if a network node has already cached the content, it recommends that to its

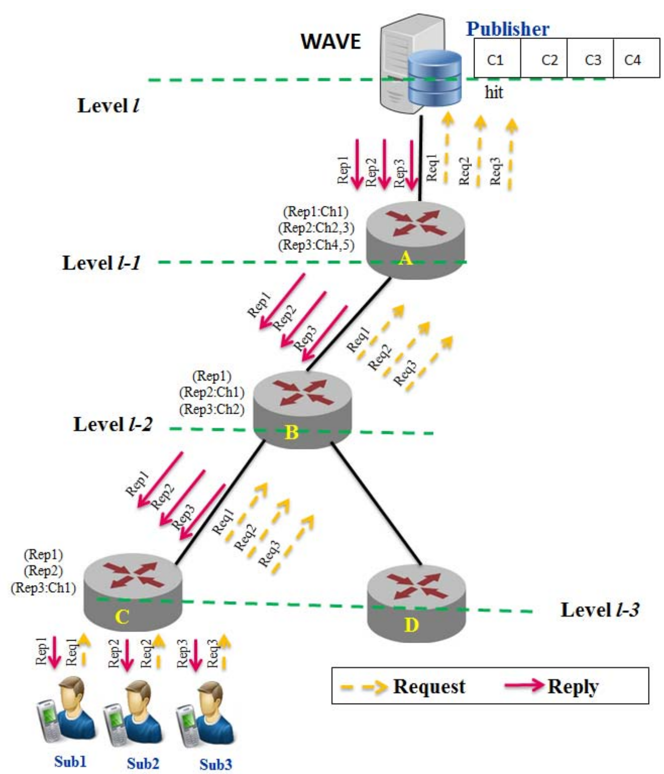


Fig. 20. Illustration of WAVE strategy.

neighbor nodes for caching (see Figure 21). In the given figure, two subscribers - *Sub1* and *Sub*3, access content *d*, which is available in the cache of router *B*, and one subscriber - *Sub2*, accesses content *c*, which is available in the cache of router *C*.

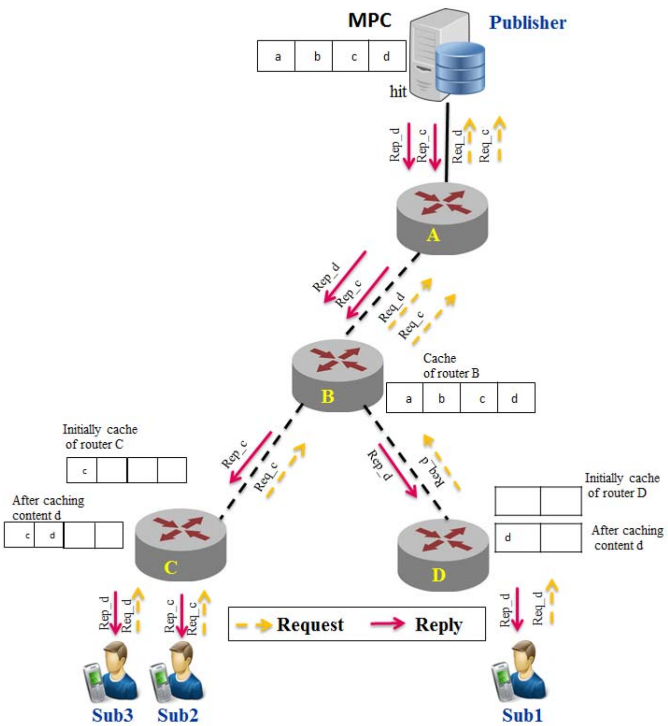


Fig. 21. MPC topology.

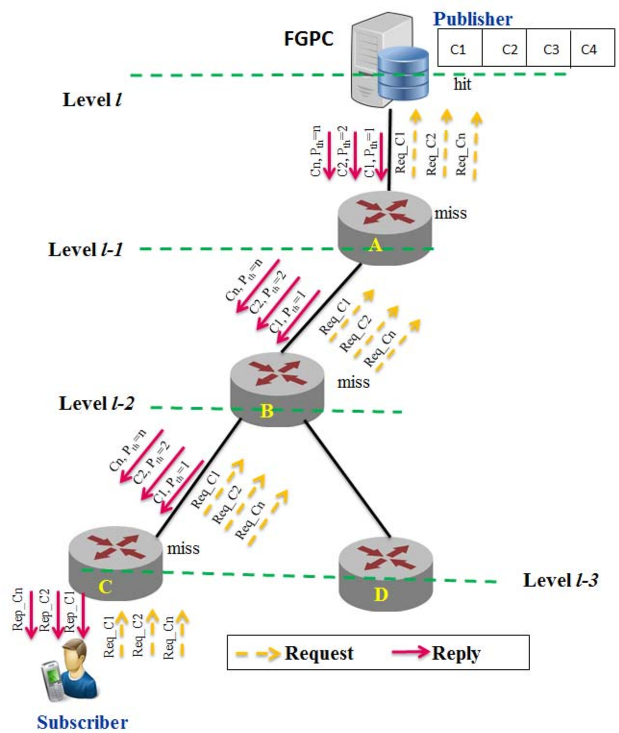


Fig. 22. FGPC topology.

In the PT, the popularity value of content d is 2 whereas the popularity value of content c is 1. It means that content d is popular and is therefore suggested to neighbor routers C and D for caching. The recommendation messages may either be accepted or denied depending on the local policies, such as resource availability. When the content popularity reduces with time after the recommendation process, the popularity count is reinitialized according to a *Reset* command.

The searching overhead is decreased in the MPC strategy, as it caches only popular contents, and thus a higher hit rate is achieved. Despite the fact that MPC achieves some higher cache hit ratio, the bandwidth and memory utilization during recommendation messages is usually high. Also, it has no policy for content eviction, so the content may simply be deleted (regardless of its popularity) upon arrival of the new one. Consequently, subsequent requests for the deleted content will be forwarded to the publisher and thus will affect the network efficiency with respect to bandwidth utilization.

6) Fine-Grained Popularity-Based Caching (FGPC): The Fine-Grained Popularity-based Caching (FGPC) [74] strategy caches all incoming contents if the CS of network node is not full. Otherwise, it stores only popular contents. Based on FGPC, a Dynamic-FGPC (D-FGPC) [74] was proposed which regularly modifies the content popularity threshold. When forwarding a content to downstream or receiving from upstream, three kinds of statistic information are updated in this strategy, i.e., content counter, content name, and time stamp. When a CS becomes full and a new content arrives, its popularity is compared with the popularity threshold value. If the value of new content is greater than the predefined threshold value, the FGPC adopts the LRU policy to cache new content in the CS, otherwise it ignores.

In Figure 22, a subscriber requests contents C_1, C_2, and C_3, respectively. For the first time as hit occurs on the publisher, the publisher replies with content C_1 and the popularity threshold (P_{th}) is set to 1 on each router on the path. Next time, if the popularity value (in the popularity table) of content C_n reaches n, the P_{th} is updated to n in all routers and one of the cached contents is evicted using LRU policy.

Although the FGPC outperforms some of the existing strategies but as it caches all incoming contents, searching overhead is increased during the replacement process. This is just because if a content is not popular then it does not contribute in the hit rate, therefore, storing it in the memory increases the computational overhead for the replacement policy. Moreover, the D-FGPC changes the threshold value according to the popularity of the content, i.e., if the popularity of content increases, the D-FGPC also increases the threshold value. For instance, if the popularity of content reaches 1,000, the threshold value will also be set as 1,000. Consequently, all those contents which have less than 1,000 popularity will never be cached, and thus the overall cache hit rate of contents is reduced.

E. IoT-Based Caching

Internet of Things (IoT) is a phenomenon where smart devices, such as mobile phones, computers, watches, or even vehicles, are connected in the environment and users can access different types of contents which are generated by these devices. The most noticeable difference between the traditional Internet and IoT is the nature of the content's request/reply.

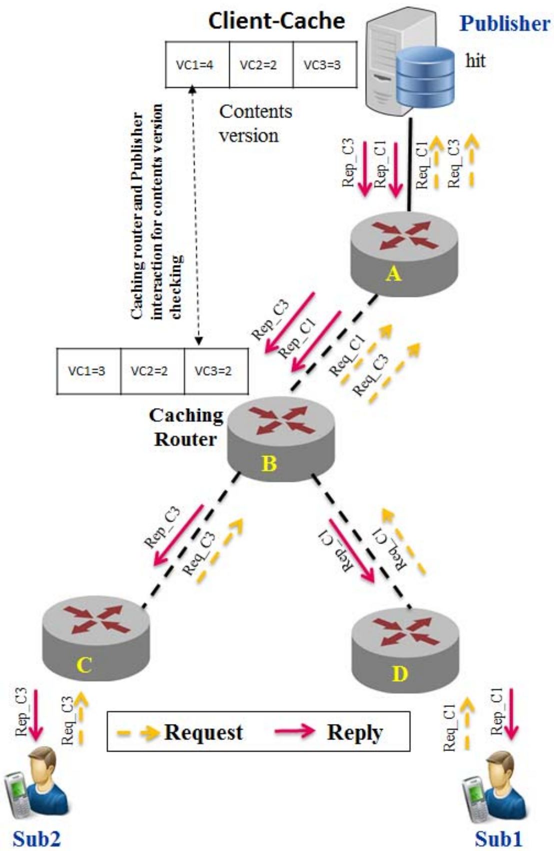


Fig. 23. Client-Cache scenario.

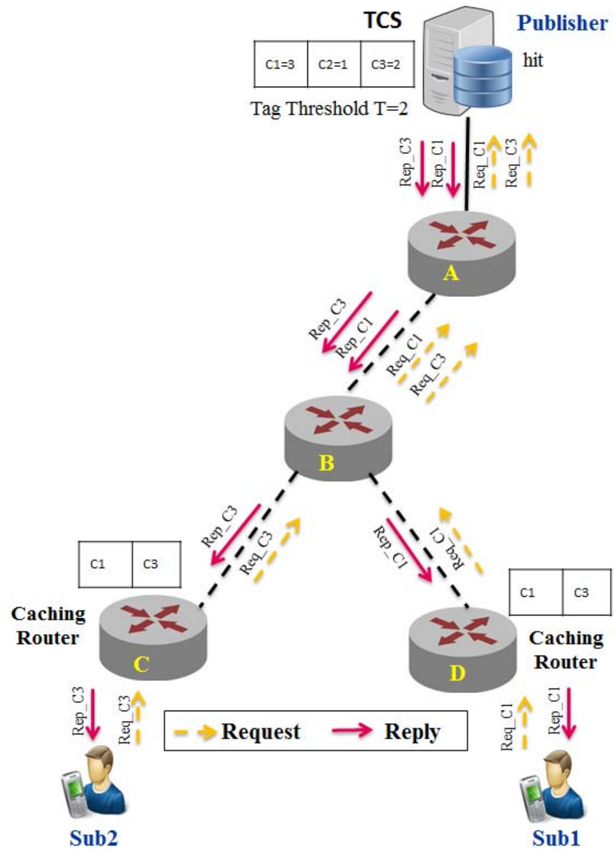


Fig. 24. TCS topology.

That is, in the existing Internet, a content is attained by sending a request for it, whereas, in the IoT the content is obtained by making the machine to initiate an action if a situation of interest is required. In other words, in the existing Internet a human generates a request for a desired content, while in the IoT the machine creates a request with the help of classified operators.

The IoT applications are well fitted in the ICN scenario, however, due to the fixed functionalities of IoT-based smart devices, the existing ICN caching strategies may not be applied in the IoT environment. For this purpose, a number of caching strategies have been developed for the IoT-based ICN scenarios so far. In this section, we discuss four popular IoT-based ICN caching strategies.

1) Client-Cache: Client-Cache [75] is a coherence caching strategy, which checks the validity of contents that are cached in the network nodes. The Client-Cache is based on three modules, i.e., caching method, caching nodes reduction, and content caching.

Firstly, the on-path caching method is selected for content caching because, unlike off-path caching, it does not require additional metadata to inform the subscriber that which node is associated with the desired publisher. Secondly, the number of caching nodes is reduced to pay attention to the most important nodes in terms of cache size and number of contents. In other words, it borrows the idea of Centrality-based caching [48], which reduces the number of caching nodes and selects only those nodes for caching which have the highest number of outgoing interfaces in the network. Thirdly, the contents are cached near the subscribers as they are connected to the edge nodes, where edge nodes are considered the most important ones.

The purpose of Client-Cache is not to have a better caching strategy but to maximize the amount of content validity, and therefore is called coherent caching strategy. The content validity is checked at the time a requested content is found in the cache. A content is considered valid if its lifetime in the publisher is greater than the lifetime of its version in the cache. This validity check involves content checking in the publisher every time a content request arrives at the local caching node.

In Figure 23, two subscribers, *Sub1* and *Sub2*, request content *C1* and *C2*, respectively. The lifetime (version) of content *C1*, *C2*, and *C3* (represented by *VC1*, *VC2*, and *VC3*) in the Publisher is 4, 3, and 2, while their lifetime in the caching router is 3, 2, and 2, respectively. It means that content *C1* and *C3* are valid as their lifetime in the Publisher is greater than their lifetime in the caching router, and content *C2* is considered invalid due to their same lifetime in both the Publisher and the caching router.

In this strategy, the validity of contents is improved but the system performance is largely degraded in terms of cache hit and hop decrement. The designers of this strategy claim that for content validity it is better to have some extra hops, which may be true, but it increases stretch ratio, content retrieval

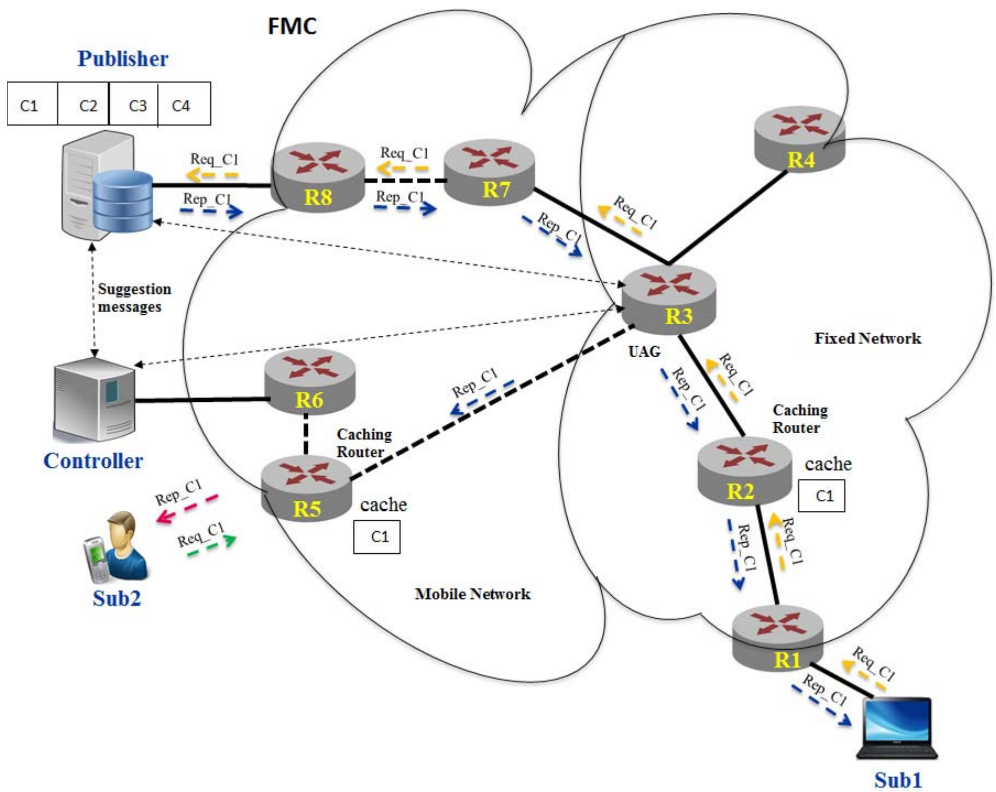


Fig. 25. Shared caching system.

delay, and eviction operations, which are considered the most prominent metrics for the ICN performance.

2) TCS: The Tag-based Caching Strategy (TCS) [76] is based on Tag Filter (TF) to lookup forwarding information and content matching. A network node maintains a list of tags for all cached contents and therefore the content store in each node identifies the locally cached tags. The tag list is used to develop a TF with the help of a hash function. When a network node receives a content object then it decides whether to cache that content object in its cache or not.

In this strategy, the tags of the contents are checked by TF in the content store. All tags of the contents are mapped and hashed to the counters that are lying in the TF of the content store. If the number of corresponding counters exceeds the threshold value, it is a signal that all tags in the local content store are popular. Consequently, each node identifies that whether a content object has reached its preferred tags or not. If the content object has reached its preferred tags then it is cached in the content store of the local node.

In Figure 24, subscribers, *Sub1* and *Sub2*, request contents *C1* and *C3* from the Publisher, respectively, where the tag threshold T is 2. As the tag value reaches the predefined threshold, the contents are cached at routers near the subscribers.

The main advantage of the TCS is to cache those contents which have the same set of tags. Also, the size of tag list in the content store is reduced and thus the performance of content matching is improved. However, if a subscriber applies the same tags in different manners then it can create ambiguity in the content matching.

3) FMC: In the Fixed and Mobile converged network (FMC) [77], an ICN-based caching scheme, known as Shared Caching System, is proposed for location independent name-based routing and fast communication. The most dominant part of this network is a functional node that works as a unified access gateway (UAG). A UAG is a gateway which is responsible for providing services for fixed and mobile users in the IoT environment. Two layers in the FMC network, i.e., network management layer and content layer, are used for making decision in the control plane. The network management layer controls traffic related functions, such as routing, while the content management layer deals with the content related matters, such as management of metadata. The UAG is utilized as a universal coordinator between the FMC network and stakeholders because it manages contents in the caching nodes of the FMC network and delivers these contents to the end users. For the content management, a cache controller is merged with the UAG, which oversees user requests and content replies from the publishers. When a network node receives a user request and it is not satisfied locally due to unavailability of the content then it is forwarded to the UAG. The UAG then forwards the request to the original publisher where the content can be satisfied. When the cache of a network node is full then every time different replacement policy is used for content replacement.

In Figure 25, a subscriber (*Sub1*) from the fixed network requests a content from the Publisher via UAG. During reply from the Publisher the UAG supplies a copy of that content to the caching routers both in the fixed and mobile networks, represented by *R2* and *R5*, respectively. Another subscriber

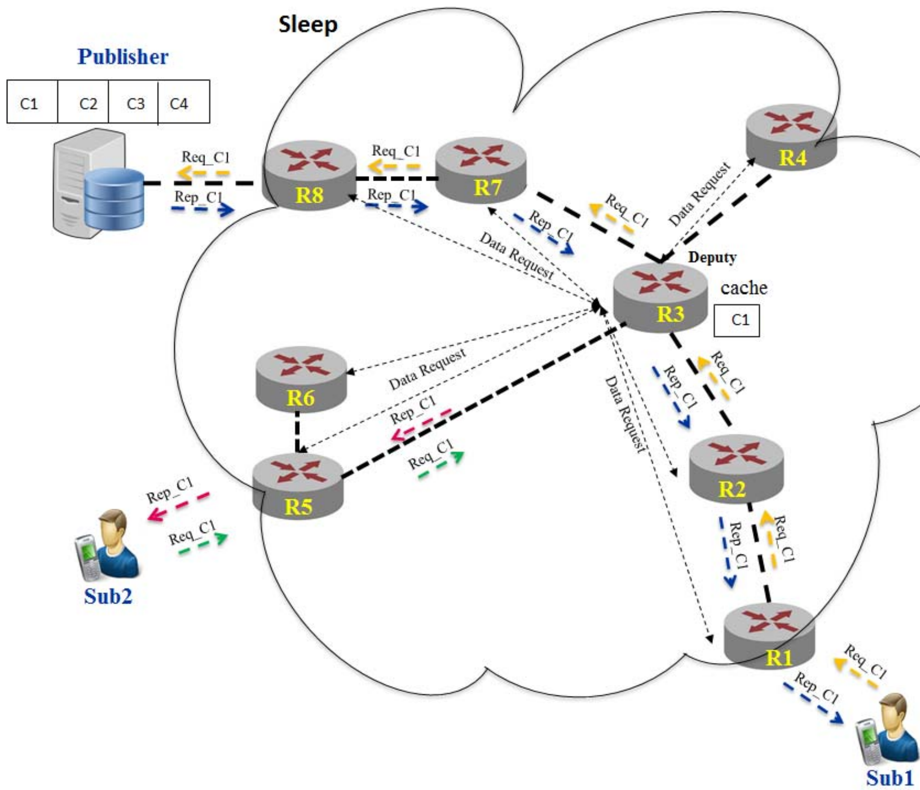


Fig. 26. Sleep-based caching strategy.

(*Sub2*) from the mobile network requests the same content and it is satisfied locally at router *R5*. The UAG and the Publisher interact with the Controller each time a content request arrives. The Controller provides caching suggestions and real-time prefetching procedures.

The Shared Caching System increases the caching efficiency at the subscriber side due to different replacement policies used in the FMC network. However, the UAG sends maximum packets to the caching nodes and hence increases the system overhead.

4) Sleep-Based Caching: The Sleep-based caching strategy [78] explains IoT in line with ICN that tries to improve energy efficiency of the network while caching the most recently accessed contents for the future use. Two things are addressed in this strategy, i.e., the first one is to schematize the sleeping mechanism that efficiently utilizes the limited energy resources, and the second one is to manage content stores of the IoT devices which have limited cache space. Moreover, the strategy also discusses three caching related modules, i.e., sleeping mode, content caching, and content replacement.

The first module, sleeping mode, is divided into two parts: uncoordinated sleep and coordinated sleep. The uncoordinated sleep mode is used in uncontrolled environment where the administrative authority is required. In this mode, every single node would decide every 'T' second its sleep and wake states. If a node is in the wake state for time 'T' and it receives a content, the node will try to cache the content with probability of 0.5. Here the preferred cache replacement method is FIFO. While in the coordinated sleep mode, the IoT nodes' sleep phase is coordinated during the network boot strap. These coordinated orders are then used to keep the nodes active for a certain period of time. However, the node can only cache a content when it is in the wake state and thus referred to as deputy node. When a node wakes up to become the deputy, it requests other nodes to send it data for caching. All network nodes can become deputies in their own allotted time. Multiple nodes can also become deputies when the amount of data exceeds the size of a single node's cache.

For the content replacement, a named based cache replacement scheme is proposed, which is known as Max Diversity Most Recent (MDMR). The content name consists of prefix and suffix, where prefix is the type of sensor and suffix is the time stamp of a content. The contents are cached and replaced on the basis of these names. The replacement method tries to replace the older content from the same producer, and also if there is a single entry per node, the oldest entry in the cache is replaced.

A prioritized prefix, known as P-MDMR, is also proposed as a replacement strategy, as some nodes perform caching and replacement based on the name to a particular cache, for example, a heat node will only cache temperatures in its content store. In case the cache is full then the contents with non-prioritized prefix would be replaced.

In Figure 26, a subscriber (*Sub1*) requests content *C1* from the Publisher. During reply the deputy node (*R3*), which is in the wake state, caches a copy of *C1* and sends data request to all network nodes to send it data, if any, for caching. The data request is sent to all network nodes because only the deputy node can cache the contents. If other subscribers need some contents, their requests are forwarded to the deputy, and the deputy fulfills their request if it has a copy of the requested content. In the given figure, *Sub2* requests content *C1* and the deputy fulfills its request as it has already cached a copy of *C1*.

With an efficient sleeping mechanism, the limited energy of IoT nodes can be managed economically in the Sleep-based caching. In addition, this strategy can extend the network lifetime and may also increase the availability of contents in the network. However, specifying the status of contents on the basis of priority increases the processing overhead.

Besides the above mentioned caching strategies, two analytical caching models have been proposed using Markov modulated rate process in [123] and Markov chains in [124] to evaluate data transfer in the CCN. Moreover, an analytical model for storage and bandwidth sharing is proposed in [125] to guide a trade-off between the limited network resources and user performance. Additionally, some experimental assessments of cache management in the ICN are presented in [126]. Some other research on caching has been proposed in [41], [56], and [127]–[138].

All the above mentioned works however do not investigate any optimum cache management strategies. In reality, placing the contents dynamically in appropriate intermediate nodes is a significant and challenging task. A summary of the presented caching strategies in terms of their contributions and limitations is expressed in Table IV.

VIII. Summary of Caching Strategies

The ICN caching is divided into different types, such as cooperative caching, non-cooperative caching, homogeneous caching, heterogeneous caching, off-path caching, and on-path caching. In cooperative caching the information related to a particular (cached) content is shared with the neighbor nodes, whereas in non-cooperative caching a network node does not advertise the information of a cached content and thus every individual node is responsible for caching decision. In homogeneous caching all network nodes on the publisher-subscriber path cache the same content, while in heterogeneous caching every node has copies of different content items as they do not cache the same contents. In on-path caching the old items are replaced with the new arrived ones after contacting the NRS, which increases the network overhead. On the other hand, on-path caching is opportunistic, where network nodes cache the content if they receive its copy. Consequently, if a new request arrives for the same content, the network node satisfies it locally without contacting the NRS. Caching copies of the contents locally may reduce the communication and computational overhead. However, it is challenging to decide that which content should be cached and what location would be suitable for caching so that the ICN performance can be enhanced in terms of cache hit rate, content redundancy and retrieval delay. For this reason, several strategies have been proposed in the literature.

Besides, the ICN caching is further classified into four categories, i.e., location-based caching, multi-level caching, single node-based caching, and popularity-based caching. In the literature, location-based caching consists of the following strategies: Hash-routing, CIC, ProbCache, Breadcrumbs, and CLS. In the Hash-routing schemes the edge node calculates the hash function upon receiving a request. If the hash function matches the content identifier of a particular router, the request is forwarded there. If the router has already cached the requested content, it replies with a copy of it, otherwise the request is forwarded to the main server. In the CIC strategy a content is divided into several chunks and those chunks are placed at more than one location in the network. The ProbCache includes TSI and TSB fields which calculate probability of the contents. The main goal of ProbCache is to place the contents near the subscriber. In the Breadcrumbs after downloading a content, a trail is created and maintained at each router for caching the routing history. In the CLS, which is the modified form of Breadcrumbs, a trail is also created for maintaining the routing history. However, in the CLS the trail is created at the time of content caching unlike Breadcrumbs which creates the trail after the content is forwarded to the underlying router.

Multi-level caching has two strategies in the literature, i.e., LCE and Prob. The LCE is also known as the default ICN caching strategy. In this strategy, a copy of the content is stored at all nodes on the publisher-subscriber path. The Prob caching strategy is a modified version of the LCE, but the main difference is such that Prob caches the object copy with probability.

Single node-based caching includes the following strategies: LCD, MCD, CATT, Betweenness-Centrality, and One-touch caching. The LCD caches a copy of the requested object at the node located immediately below the node where hit occurs. The MCD is similar to LCD in functionality, but if hit occurs at a node then the copy is deleted from that particular node and is shifted to the underlying cache. In the CATT strategy, on the edges of an AS some CATT nodes are deployed as caching nodes and the contents are cached at a single location on the publisher-subscriber path. The Betweenness-Centrality caches contents at a node which has the maximum number of routes traversing it. One-touch caching scheme stores contents at the server based on their access time. In case of memory overflow, the contents are randomly replaced with the new arrived ones.

Popularity-based caching comprises OCPCP, NCCM, TLRU, WAVE, MPC, and FGPC strategies. The OCPCP caches contents based on their popularity value which is calculated during content caching. The NCCM scheme is based on SDN and jointly considers content routing and caching through LNC. The TLRU is a content eviction policy where a local node calculates time stamps of the arrived contents. If the average request time of the arrived content is smaller than the time stamp of the cached contents then it is recommended for caching. The WAVE organizes chunks on the basis of their access time. The upstream router in the WAVE strategy decides that which content should be cached at the downstream router. The MPC also caches contents based on their popularity value, however, it introduces a popularity table for storing content name and its popularity value. In the MPC, the neighbor nodes are suggested to cache the content when its popularity value reaches the popularity threshold. The FGPC is the modification of MPC that also includes a popularity table for caching both

TABLE IV
SUMMARY OF CHARACTERISTICS OF THE ICN CACHING STRATEGIES

Caching Strategy	Contributions	Limitations
Hash-routing [61]	Reducing caching redundancy, as the hash function indicates that the content is cached wherever it is found.	Increasing the possibility of intradomain link load in both multicast hash-routing and symmetric approaches.
CIC [62]	Improving the caching diversity by storing different kinds of chunks and boosting the average response time.	Increasing computational overhead with respect to searching time if some chunks need to be deleted for creating space for a new arrived content.
LCE [63]	Increasing hit ratio and reducing content retrieval delay and stretch.	Increasing redundancy by making the cache of all intermediate nodes full.
Prob [63]	Reducing redundancy by caching a copy of the requested content on the path from the hit location to the requesting host.	Minimizing hit ratio if the cache of all intermediate nodes become full and no space is available for the new arriving contents.
LCD [63]	Reducing redundancy by caching a copy of the requested content at the node which is located immediately below the node where hit occurs.	Overwhelming the cache of nodes on the link and utilizing maximum bandwidth.
MCD [64]	Reducing the number of copies for the same object between the server and the requesting host.	Increasing content request delay because if a content is evicted from the cache and it is requested again then it can only be replied from the server.
ProbCache [47]	Maximizing the number of different content items stored along a delivery path to reduce redundancy between network caches.	Decreasing *hop decrement* because it frequently replaces contents at nodes near the user(s).
Breadcrumbs [65]	Maintaining routing history by creating a trail at each router that helps to reduce the server overhead.	Maximizing retrieval delay during downloading because if the content is evicted at each router, the request may suffer missing downstream.
CLS [66]	Maintaining routing history by creating a trail at each router, which guarantees that the CLS saves cache space as it deletes the trail on time.	Maintaining redundancy and increasing content delay (as it cannot ensure the placement of chunk at the nearest node to the user).
OCPCP [67]	Minimizing content redundancy and retrieval delay and improving the network throughput.	Increasing the chances of filling router's cache in a network very rapidly and hence reducing the cache hit ratio as excessive content replacement may happen.
CATT [68]	Minimizing content retrieval delay and redundancy by caching contents only at the edge routers of ASs.	Reducing cache hit ratio if the cache of edge routers overflow and the new contents need to replace the cached ones.
Betweenness-Centrality [48]	Minimizing redundancy by storing content objects only once along the path.	Lacking mobility, i.e., difficult for a router to find the value of its betweenness centrality if it is deployed on mobile ad-hoc networks.
One-touch caching [69]	Reducing computational complexity by preserving the most recently requested contents in the server.	Utilizing maximum bandwidth during request to and/or reply from the servers (as all requests are served by content servers).
NCCM [70]	Minimizing the network bandwidth cost by reporting content request statistics periodically with a fixed time period by all CRs.	Facing problems in the implementation due to lacking support for network coding.
TLRU [71]	Caching arriving content if the average request time is smaller than the time stamps of the stored contents and therefore increasing cache hit rate.	Boosting the chance of filling cache of content routers in a network very rapidly and hence reducing its effectiveness for content eviction.
WAVE [72]	Increasing the cache hit ratio because of content popularity.	Decreasing the performance as it cannot eliminate caching redundancy.
MPC [73]	Storing popular contents which helps to improve cache hit ratio and decrease content redundancy.	Consuming maximum bandwidth and memory during suggestion messages.
FGPC [74]	Achieving higher hit rate and fast convergence speed till the moment the CS has a storing capacity.	Reducing computational complexity as the threshold value is dynamically modified according to the content popularity.
Client-Cache [75]	Improving the validity of contents.	Degrading the system performance in terms of cache hit and hop decrement.
TCS [76]	Improving the forwarding efficiency with the help of tags.	Creating ambiguity if the same tag is applied to different contents.
FMC [77]	Increasing the caching efficiency at the subscriber side.	Increasing the system overhead because the UAG sends maximum packets to the caching nodes.
Sleep-based Caching [78]	Managing energy of the IoT devices efficiently.	Increasing computational overhead due to content specification on the basis of priority.

the content name and its popularity value. However, the threshold in the FGPC is regularly modified based on the content popularity value.

IoT-based caching consists of four strategies, i.e., Client-Cache, FMC, TCS, and Sleep-based caching. The Client-Cache strategy is used to maximize the amount of content

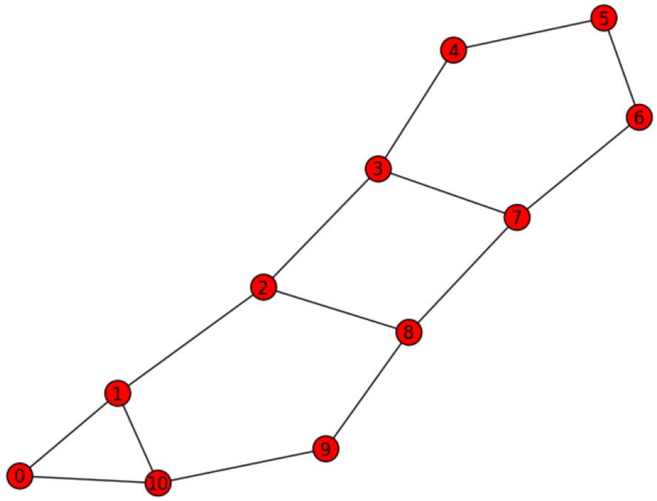


Fig. 27. Abeline topology.

validity. The validity of contents is measured on the basis of their lifetime. That is, if the lifetime of a content in the publisher is greater than the lifetime of its version in the local cache, then it is considered a valid content. In the TCS, the contents are cached on the basis of tags. A content is stored in the local node's cache if its tag value has reached the preferred number. The FMC caches contents with the help of content management layer and the UAG. The UAG works as a system coordinator and delivers the requested contents to the end users. In the Sleep-based caching, a content is cached based on its prefix and suffix names, where prefix is the type of sensor and suffix is the time stamp of the content.

IX. Performance Evaluation

For the evaluation of the studied strategies, SocialCCNSim [139], [140] was used, which is a simulator specifically designed for testing the performance of ICN caching. The SocialCCNSim downloads contents from a Facebook social network topology [50]–[52], which was designed by a research group at Stanford University for research purposes. The topology consists of 4,039 users, 88,234 friends, and each user counts 44 relationships [139].

A. Simulation Environment

In our simulations, LRU cache replacement policy was used for the content replacement from all ICN nodes. The LRU policy was chosen because most of the existing strategies use this policy for content replacement. In SocialCCNSim, for modeling content requests, the Zipf distribution function [141]–[143] was used, which counts content probability, where skewness of the distribution is represented by α. We have used Abilene topology (see Figure 27) - a popular ISP-level topology that has 11 stations as nodes, for evaluating these strategies (see Table V). Abilene topology is also known as Internet2 topology [144] and was designed with the aim to exchange the huge amount of information across nodes. This topology was chosen because its building is perfect for testing and comparing the simulation parameters.

TABLE V
Simulation Scenario

Parameter	Description
Cache Size	1GB - 10 GB
Catalog Size	10^8 elements
α Parameter	0.8
Topology	Abilene
Social Network Topology	Facebook [50], [51], [52]
Simulator	SocialCCNSim [139], [140]
No. of Simulation	10 runs

In our simulations, the Zipf probability distribution was used as popularity model with the α parameter being 0.8; the cache size (which specifies the available space in every node for temporally storing content objects) varied from 1GB to 10GB; and the catalog (which represents the total number of contents in a network) was 10^8 elements. The content or element means file and has a size of 10MB each. The α parameter is chosen as 0.8 because this represents Web contents [142]. Therefore, it is better to test performance of the strategies in a network environment where the Internet traffic is too high. The scenario was simulated for 10 numbers of runs and the average values were taken for cache hit, stretch ratio, and eviction operations, as shown in Figures 28, 29, and 30, respectively. The simulations were performed for 10 times, where each running time was 24 hours. The reason for 10 times running was such that the evaluated cache size was 1GB to 10GB, therefore, for each cache size the simulation was run and the average results were collected for cache hit, stretch ratio, and eviction operations.

B. Simulation Results

In this section we present the simulation results of all strategies except One-touch caching because, as mentioned earlier, One-touch is designed for the server, which replaces contents on the origin server and is not suitable for local caching like other strategies. In the given figures, the X-axis shows 1GB to 10GB cache size, which is represented by 1 through 10. It can be seen that all strategies produced different results in different scenarios, which are discussed in the following subsections.

1) Cache Hit: Cache Hit is the most significant metric in evaluating the performance of ICN and refers to the portion of content requests satisfied by caches implemented within the network. This metric shows the potential of the caching strategy to moderate the number of redundant content copies.

In the location-based caching strategies, as shown in Figure 28(a), the CLS strategy achieved the highest cache hit rate, whereas the Breadcrumbs was the lowest in the cache hit. The reason behind this result is such that, as discussed in Sections VII-A4 and VII-A5, both Breadcrumbs and CLS create trails. However, the CLS trail is maintained at the time of chunk caching, which helps the CLS to find the chunk during the downstream searching. In contrast, the Breadcrumbs trail is generated at the time the content is pushed down, therefore, if the content is evicted at a router, the request may suffer missing downstreams.

In the single node-based caching strategies (Figure 28(b)), LCD achieved the highest cache hit in comparison with MCD, CATT, and Betweenness-Centrality (represented by

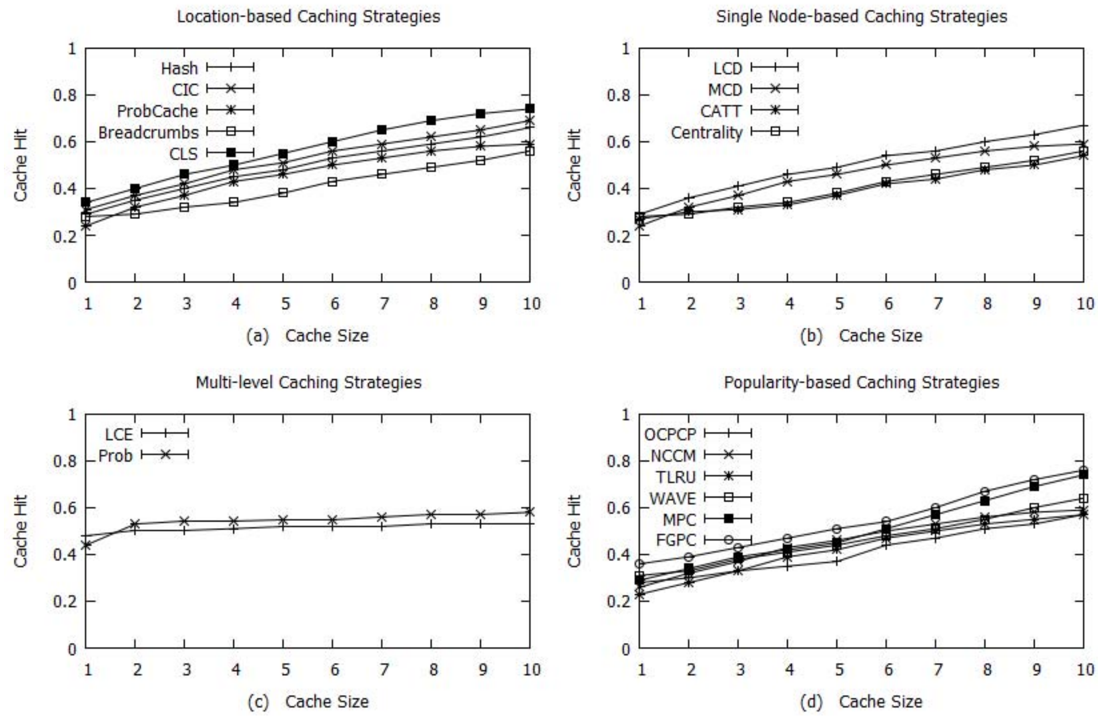


Fig. 28. Cache hit rate, where cache size is 1-10GB.

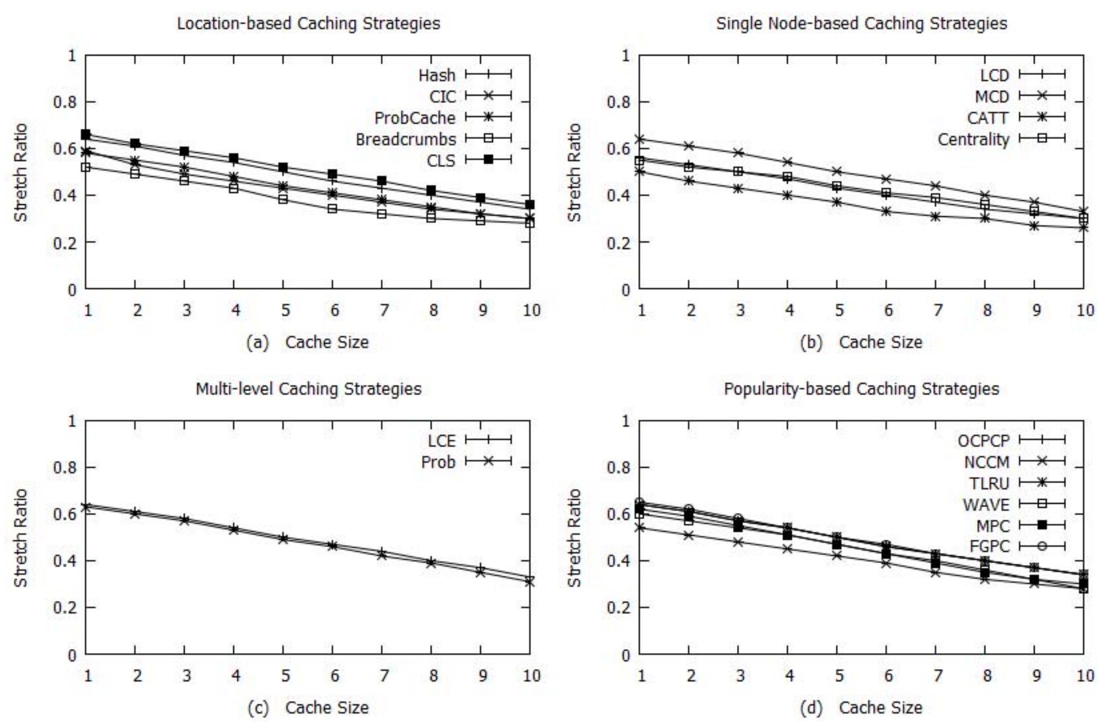


Fig. 29. Stretch ratio with cache size 1-10GB.

Centrality). The reason behind the highest cache hit of LCD is due to the fact that, unlike other strategies, it reduces the caching redundancy by placing the most popular content near the subscriber(s).

In the multi-level caching strategies, which have only two strategies in our simulations, i.e., LCE and Prob, there is no such difference in the results because Prob is the modified version of LCE. However, a little result supremacy of Prob over LCE is that Prob caches contents with probability.

In the popularity-based caching strategies (Figure 28(d)), FGPC achieved the highest result with low cache sizes, but with the increase of cache size (from 1GB to 10GB) MPC and

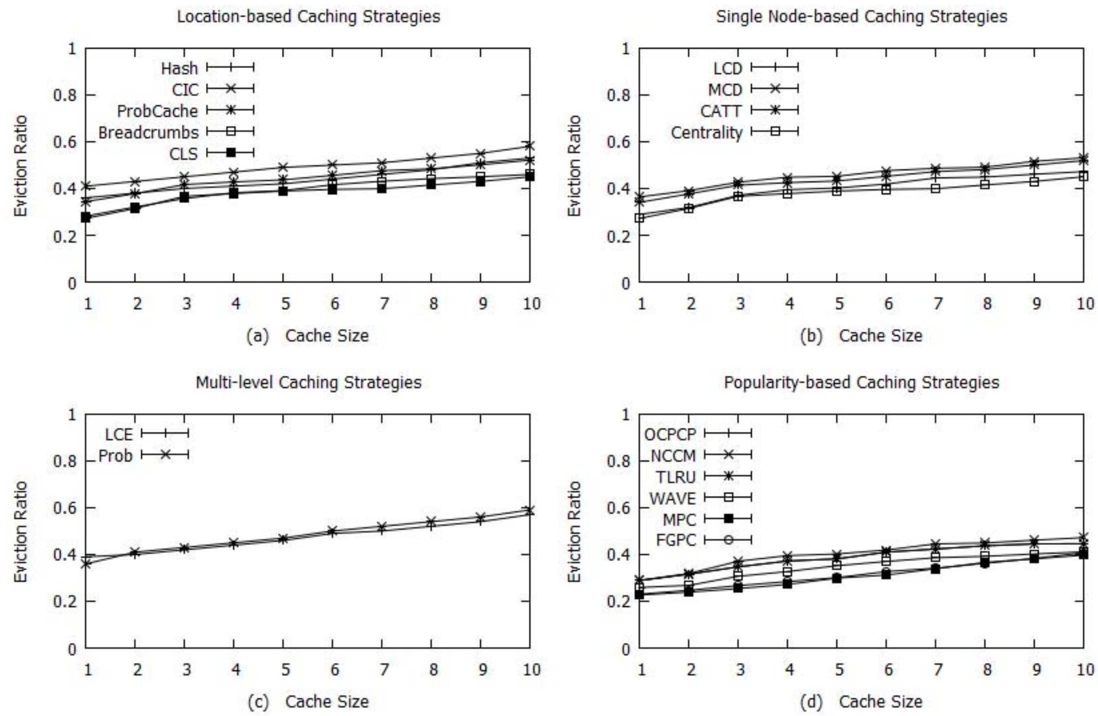


Fig. 30. Eviction operations with cache size 1-10GB.

FGPC achieved almost the same result. As compared to other popularity-based caching strategies, MPC caches the most popular contents and thus its result is higher than the other strategies except that of the FGPC. The FGPC has the highest results (with low cache sizes) because it updates the popularity threshold regularly and therefore may cache the most popular contents.

In the IoT-based caching strategies (Figure 31(a)), TCS achieved the highest result as compared to Sleep-based caching (represented by Sleep), Shared Caching System (represented by FMC), and Client-Cache (Fig. 31(b)). This is due to the fact that TCS caches contents with the same set of tags which help in increasing content popularity, and thus popular contents help in increasing the cache hit rate. FMC was the lowest in cache hit with 56% because Wi-Fi and mobile networks are administered separately, therefore, their controllers do not have the same information about the cached contents.

2) Stretch: Stretch refers to the number of hops a content request travels from a client to the server to find a content in the network. Stretch depends on the *hop decrement*, that is, if the *hop decrement* is high, the content delay is reduced and vice versa. The high *hop decrement* means that the content is cached close to the subscriber. As a result, it will not take long time in downloading.

In the location-based caching strategies (see Figure 29(a)), stretch ratio was the lowest (the lowest being the best) in the Breadcrumbs strategy with low cache sizes. However, with the increase of cache size from 1GB to 10GB, ProbCache achieved almost the same result as that of the Breadcrumbs. This is because both ProbCache and Breadcrumbs drag the content to place it near the subscriber(s). As for the Hash and CIC strategies, the contents are cached at different off-path nodes in the forms of chunks, the stretch ratio is a bit higher. The CLS strategy performed worst in the stretch because, as discussed in Section VII, according to the CLS trail, the in-network nodes are unable to judge whether the cached copy is closer than the server or not. Thus, the CLS cannot guarantee to cache the chunks closer to the subscriber.

In the single node-based caching strategies (Figure 29(b)), CATT achieved the lowest stretch ratio in comparison with LCD, MCD, and Centrality. The reason behind CATT's supremacy is that, according to the nature of CATT, it tries to cache the contents near the user.

In the multi-level caching strategies, same as the cache hit, the stretch ratio is also similar in the Prob and LCE (Figure 29(c)), because both caches the contents at every on-path node. Generally, it seems that caching content at every network node can ensure that the requested content may be found near the user. Unfortunately, this is not true as the chances to fill the caches of network nodes are very high due to the unpredictable growth in the number of Internet users. Thus, upon filling the node's cache, the new arrived contents may replace the cached ones in the LCE and Prob, and therefore the requested content may not be found locally.

In the popularity-based caching strategies, there was no big difference in the results except for NCCM, which achieved the lowest stretch with low cache sizes. However, when the cache size was increased to 10GB, all strategies produced almost the same stretch, as shown in Figure 29(d).

In the IoT-based caching strategies (Figure 31(c) and 31(d)), the stretch ratio in Sleep, Client-Cache, and FMC is almost the same, but TCS is lowest in this ratio because if the same

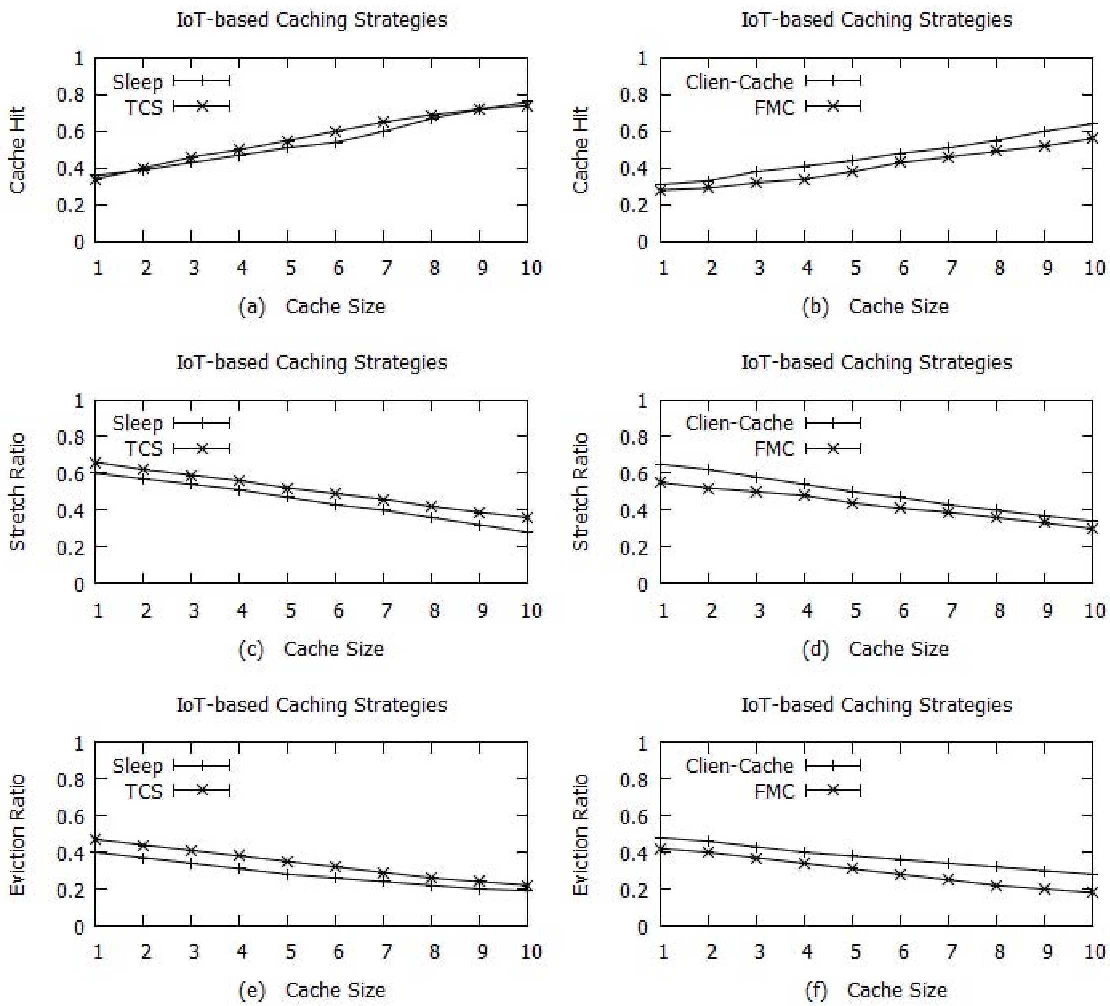


Fig. 31. Hit, stretch, and eviction ratios with cache size 1-10GB.

tags are applied to different contents then it creates ambiguity in content matching. On the other hand, the reason of identical result of the remaining three strategies is due to mobility of the IoT devices, where the subscribers move from one point of access to the other. Thus, it is difficult to decide that which device will be connected to which router in a specific period of time.

3) Eviction: Eviction is also one of the popular metrics for the performance evaluation of ICN. When the cache of a network node is full and a new content arrives then one of the cached contents is evicted to accommodate the new arrived one. At the time of excessive eviction, the overall network throughput is affected in terms of content retrieval delay and hit ratio. In other words, if a stored content is evicted and a new request arrives for it, then the request is forwarded to the server to satisfy the need. As a result, the delay in content retrieval increases and the cache hit decreases.

For the eviction ratio, the CLS and Breadcrumbs strategies produced almost the same results, which were lower than those of the other location-based caching strategies, as shown in Figure 30(a). In the single node-based caching strategies, Centrality and LCD produced somehow the same results with both low and high cache sizes (Figure 30(b)). In the multi-level caching strategies, again no big difference was seen in the eviction ratio in both LCE and Prob. In the popularity-based caching strategies (Figure 30(d)), MPC and FGPC generated the lowest stretch ratio. This is because MPC caches content based on the popularity table, i.e., when the content popularity reaches a predefined threshold value in the popularity table then it is recommended for caching. Thus, the cache of a network node does not fill rapidly and hence the eviction is not needed until the cache becomes full. Similarly, FGPC updates the threshold value periodically and therefore the most popular content is cached.

In the IoT-based caching strategies (Figure 31(e) and 31(f)), the eviction ratio of Sleep, TCS, and FMC is also somehow similar as their stretch ratio, but Client-Cache has the highest eviction ratio with 26% because it takes some extra hops for content validity. The reason behind similar results of the other strategies is such that mobility is involved in the IoT environment. Accordingly, in mobile ad hoc networks it is hard to know that which particular node will cache a content and from which node a content needs to be evicted.

X. Challenges and Future Research Directions

Although ICN caching has made remarkable improvement, there are several yet to solve challenges. In this section we address some unresolved challenges that need attention from the ICN research community.

A. Web Application Popularity

Content placement and replacement mechanisms have been extensively thought-out at the application level, for the most part with regards to Web applications. It has been submitted in a paper [145] that the advantages from the widespread utilization of ICN caching will not have a good impact on the network throughput. The authors raised this point based on the previous studies. However, those analyses were generally based on research conducted more than a decade ago [146]. Further research on existing Internet design could reveal extra insight into the significance of data today and subsequently to the achievable preferences from widespread caching. For example, a recent study has demonstrated that, in the last decade, caching performance at the application level has been effectively changed by Web information popularity [1], [147].

B. Traffic Type and Caching Space

When content items are cached inside the network, various traffic types will compete for the same storing space, therefore, it becomes critical for the network to manage cache space. Recent studies on cache management have shown that intelligent and flexible strategies can significantly improve the caching performance [47], [48], [126].

Furthermore, on-path caching is a simple technique to content placement as they move from source to destination. The communication and computational complexity can be reduced using on-path caching but, conversely, the chances of hitting cached content items might also be reduced. In addition, mechanisms for storing content items inside the network open up the possibility of in-network cache management, routing and forwarding. For example, cache locations, indications of cache contention and cache-ability of information can affect routing decisions [1].

C. Staleness Detection

Staleness detection of cached NDOs can be another challenging task in ICN caching. As copies of NDOs are largely distributed in in-network caches, ICN must have the capability of providing staleness verification procedure for synchronization of NDOs placed at their publishers and in-network caching nodes. To handle this issue, two types of approaches, namely direct and indirect approaches, can be considered. The direct approach is suitable for some NDOs because in this approach each cache searches specific information in the name of NDOs, for example, time stamp which directly specifies its staleness. In the indirect approach, the provider of the cached NDO is consulted about its staleness before providing it. This approach is well applicable to those NDOs for which it is difficult to set their expiring time in advance, for example, a webpage that includes the main text (which remains the same) and the interactive portion (which does not need regular revision) such as ads and comments [89].

D. Content Deduplication

Content deduplication is also a challenging task in ICN that may create issue in accessing data objects [148]. That is, if a user requests a content and is cached at a network node (for example A) with a name (say M), and the same content is requested by another user and is cached with a different name (say N) at the same node (i.e., A). Then the same content with a different name is cached at one node, which increases content redundancy. Similarly, if the chunks of a multimedia content are cached at different locations, it may raise issues in accessing these chunks as all chunks would have different names. On the other hand, if the names of all chunks are the same, then it may introduce severe delay, extra buffering periods, and poor quality [59].

E. Content Redundancy

Redundancy is another critical issue in ICN, for instance, frequently replicating content items in all network nodes results in redundant replicas of the same items. In this regard, previous research tries to reduce redundancy of stored content items. However, the issue of storing analogous content items in on-path caches still remains open.

F. Mobility

Mobility is also considered a very important module because of the huge number of mobile users. Unfortunately, this module received very little attention from the ICN research community as most of the proposed caching strategies are designed for fixed wireless networks. But in reality, mobile traffic is gaining enormous popularity [149] and according to Cisco Visual Networking Index [14] it will reach 18 times more than the current mobile traffic by 2021. ICN mobility may be categorized in three different modes, i.e., user mobility, content mobility, and network mobility. In the user mobility, the users move during content requests and they cannot be associated with a specific location. In the content mobility, the location of a certain content change continuously, i.e., if a user downloads a particular content and it is cached somewhere in the network, the next user requests the same content from another location and (according to most of the existing caching strategies) it is cached at some other location. Hence, most of the requests associated with that content are disturbed because the routing information needs to be updated. In the network mobility, when a user in a network, such as Vehicular Ad hoc Network (VANET) or Mobile Ad hoc Network (MANET), change its location, this movement interrupts all the associated connections. In the network mobility, a lot of information (related to network connections) is required to be updated, which further creates more difficulties for name resolution as well as routing.

XI. Conclusion

ICN is acceptably a dominant nominee for the future Internet architecture. In this article, we have endeavored to give a deep survey of the ICN research environment. As an initial step, we presented the general idea of ICN, and then familiarized the typical CCN-based ICN approach. After that, we presented the issues of ICN in-network caching, with a major bloom of existing cache management strategies taking place in the last decade in line with their contributions and limitations. We also compared the existing strategies through extensive simulations in terms of cache hit rate, stretch ratio, and eviction operations. It is argued that eliminating the content redundancy (i.e., caching popular contents), increasing hop decrement, and reducing content retrieval delay are the actual reasons for improving the ICN network throughput with respect to cache hit, stretch, and eviction ratios. Furthermore, multimedia applications also play an important role in the ICN environment. This argument is strengthened by the simulation results, where popularity-based caching strategies performed better than the others. And the most popular contents are multimedia applications as revealed by the Cisco Visual Networking Index that multimedia traffic will reach 82 percent of the current Internet traffic in 2021. This led to a discussion of the research challenges and open issues for ICN in-network caching which, though essential, have not received much interest from the majority of the research community.

REFERENCES

[1] 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.

[2] A. Feldmann, "Internet clean-slate design: What and why?" *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 3, pp. 59–64, 2007.

[3] M. Handley, "Why the Internet only just works," *BT Technol. J.*, vol. 24, no. 3, pp. 119–129, 2006. [Online]. Available: http://www0.cs.ucl.ac.uk/staff/M.Handley/papers/only-just-works.pdf

[4] N. Fotiou, "Information—Centric networking: Security requirements and solutions," Ph.D. dissertation, School Inf. Sci. Technol., Dept. Comput. Sci., Ahens Univ. Econ. Bus., Athens, Greece, 2014. [Online]. Available: http://phdtheses.ekt.gr/eadd/handle/10442/34673

[5] V. Jacobson, "A new way to look at networking," Google Tech Talk, Aug. 2006, accessed: Sep. 24, 2016. [Online]. Available: https://www.youtube.com/watch?feature=player_embedded&v=8Z685OF-PS8

[6] *Information-Centric Networking Research Group (ICNRG)*. Accessed: Dec. 19, 2016. [Online]. Available: http://trac.tools.ietf.org/group/irtf/trac/wiki/icnrg

[7] D. Trossen, M. Sarela, and K. Sollins, "Arguments for an information-centric internetworking architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 2, pp. 26–33, 2010.

[8] *What Happens in an Internet Minute*, Intel, Santa Clara, CA, USA, accessed: Feb. 2, 2017. [Online]. Available: http://scoop.intel.com/what-happens-in-an-internet-minute/

[9] J. Rexford and C. Dovrolis, "Future Internet architecture: Clean-slate versus evolutionary research," *Commun. ACM*, vol. 53, no. 9, pp. 36–40, 2010.

[10] H. Li, H. Nakazato, A. Detti, and N. B. Melazzi, "Popularity proportional cache size allocation policy for video delivery on CCN," in *Proc. Eur. Conf. IEEE Netw. Commun. (EuCNC)*, Paris, France, 2015, pp. 434–438.

[11] F. Malabocchia *et al.*, "Using information centric networking for mobile devices cooperation at the network edge," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, Glasgow, U.K., 2015, pp. 1–6.

[12] H. Nakazato *et al.*, "On-path resolver architecture for mobility support in information centric networking," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, San Diego, CA, USA, 2015, pp. 1–6.

[13] S. Carew, "Users complain, AT&T blames data tsunami," Reuters Media File, Feb, 2012, accessed: Sep. 10, 2016. [Online]. Available: http://blogs.reuters.com/mediafile/2012/02/14/users-complain-att-blames-data-tsunami/

[14] "Cisco visual networking index: Forecast and methodology: 2016–2021," Rep., Sep. 2017, accessed: Oct. 18, 2017. [Online] Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html

[15] "The zettabyte era: Trends and analysis," Rep., Jun. 2017, accessed: Jul. 2, 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html

[16] *What Happens in an Internet Minute in 2016*, Intel, Santa Clara, CA, USA, accessed: Feb. 1, 2017. [Online]. Available: http://www.visualcapitalist.com/what-happens-internet-minute-2016/

[17] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking (draft)," in *Proc. Dagstuhl Seminar Schloss Dagstuhl Leibniz Zentrum für Informatik*, 2011, pp 1–26.

[18] *World Wide Web Consortium, Internet Live Stats*. Accessed: Dec. 22, 2016. [Online]. Available: http://www.internetlivestats.com/

[19] Y. Jin and Y. Wen, "PAINT: Partial in-network transcoding for adaptive streaming in information centric network," in *Proc. IEEE 22nd Int. Symp. Qual. Service (IWQoS)*, Hong Kong, 2014, pp. 208–217.

[20] A. Seema, L. Schwoebel, T. Shah, J. Morgan, and M. Reisslein, "WVSNP-DASH: Name-based segmented video streaming," *IEEE Trans. Broadcast.*, vol. 61, no. 3, pp. 346–355, Sep. 2015.

[21] 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.

[22] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, "A survey on content-oriented networking for efficient content delivery," *IEEE Commun. Mag.*, vol. 49, no. 3, pp. 121–127, Mar. 2011.

[23] A. Detti, B. Ricci, and N. Blefari-Melazzi, "Mobile peer-to-peer video streaming over information-centric networks," *Comput. Netw.*, vol. 81, pp. 272–288, Apr. 2015.

[24] 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.

[25] T. Koponen *et al.*, "A data-oriented (and beyond) network architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 181–192, 2007.

[26] *NSF Named Data Networking Project*. Accessed: Mar. 5, 2017. [Online]. Available: www.named-data.org/

[27] A. Karami and M. Guerrero-Zapata, "An ANFIS-based cache replacement method for mitigating cache pollution attacks in named data networking," *Comput. Netw.*, vol. 80, pp. 51–65, Apr. 2015.

[28] J. H. Mun and H. Lim, "Cache sharing using a bloom filter in named data networking," in *Proc. Symp. Archit. Netw. Commun. Syst.*, Santa Clara, CA, USA, 2016, pp. 127–128.

[29] *NSF Mobility First Project*. Accessed: Jan. 22, 2016. [Online]. Available: http://mobilityfirst.winlab.rutgers.edu/

[30] *CCN Project*. Accessed: Nov. 2, 2017. [Online]. Available: https://wiki.fd.io/view/Cicn

[31] P. Jokela, A. Zahemszky, C. E. Rothenberg, S. Arianfar, and P. Nikander, "LIPSIN: Line speed publish/subscribe inter-networking," in *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 195–206, 2009.

[32] B. Ahlgren *et al. Second NetInf Architecture Description, The FP7 4WARD Project*. Accessed: Aug. 22, 2016. [Online]. Available: http://www.4ward-project.eu/index9d96.html?s=Deliverables

[33] B. Ahlgren *et al.*, "Design considerations for a network of information," in *Proc. ACM CoNEXT Conf.*, Madrid, Spain, 2008, p. 66.

[34] *FP7 Publish-Subscribe Internet Technology: PURSUIT Project*. Accessed: Mar. 13, 2017. [Online]. Available: http://www.fp7-pursuit.eu/PursuitWeb/

[35] *FP7 CONVERGENCE Project*. Accessed: Mar. 11, 2017. [Online]. Available: http://www.ict-convergence.eu/

[36] *FP7 Scalable and Adaptive Internet Solutions: SAIL Project*. Accessed: Mar. 11, 2017. [Online]. Available: http://www.sail-project.eu/

[37] *FP7 COntent Mediator Architecture for Content-Aware nETworks: COMET Project*. Accessed: Mar. 11, 2017. [Online]. Available: http://www.comet-project.org/

[38] V. Jacobson *et al.*, "Networking named content," in *Proc. 5th Int. Conf. Emerg. Netw. Exp. Technol.*, Rome, Italy, 2009, pp. 1–12.

[39] K. Pentikousis *et al.* (2015). *Information-Centric Networking: Baseline Scenarios*. Accessed: Feb. 1, 2016. [Online]. Available: http://www.rfc-editor.org/info/rfc7476

[40] J. Seedorf, M. Arumaithurai, A. Tagami, K. Ramakrishnan, and N. Blefari-Melazzi, *Research Directions for Using ICN in Disaster Scenarios*, in Internet-draft, Jul. 2017, accessed: Aug. 15, 2017. [Online]. Available: https://tools.ietf.org/html/draft-irtf-icnrg-disaster-02

[41] S. Hassan, I. U. Din, A. Habbal, and N. H. Zakaria, "A popularity based caching strategy for the future Internet," in *Proc. ITU Kaleidoscope ICTs Sustain. World (ITU WT)*, Bangkok, Thailand, 2016, pp. 1–8.

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

[43] "A taxonomy of caching approaches in information-centric network architectures," *Elsevier J.*, 2013, accessed: Jul. 4, 2017. [Online]. Available: https://scss.tcd.ie/publications/tech-reports/reports.15/TCD-CS-2015-02.pdf

[44] 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.

[45] 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.

[46] 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.

[47] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *Proc. 2nd Ed. ICN Workshop Inf. Centric Netw.*, Helsinki, Finland, 2012, pp. 55–60.

[48] 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.

[49] J. Ren *et al.*, "MAGIC: A distributed MAx-gain in-network caching strategy in information-centric networks," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, 2014, pp. 470–475.

[50] J. Leskovec and A. Krevl. (Jun. 2014). *SNAP Datasets: Stanford Large Network Dataset Collection*. accessed: Jun. 22, 2016. [Online]. Available: http://snap.stanford.edu/data

[51] J. Leskovec and J. J. Mcauley, "Learning to discover social circles in ego networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 539–547. [Online]. Available: http://cs.stanford.edu/people/jure/pubs/circles-nips12.pdf

[52] *Stanford Network Analysis Project (SNAP)*, Accessed: Jan. 2, 2016. [Online]. Available: http://snap.stanford.edu/data/egonets-Facebook.html

[53] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A survey of energy-efficient caching in information-centric networking," *IEEE Commun. Mag.*, vol. 52, no. 11, pp. 122–129, Nov. 2014.

[54] 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.

[55] J. Wang, R. Wakikawa, and L. Zhang, "DMND: Collecting data from mobiles using named data," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Jersey City, NJ, USA, 2010, pp. 49–56.

[56] N. B. Melazzi, A. Detti, M. Arumaithurai, and K. K. Ramakrishnan, "Internames: A name-to-name principle for the future Internet," in *Proc. 10th Int. Conf. Heterogeneous Netw. Qual. Rel. Security Robustness (QShine)*, 2014, pp. 146–151.

[57] A. Detti, B. Ricci, and N. Blefari-Melazzi, "Supporting mobile applications with information centric networking: The case of P2Plive adaptive video streaming," in *Proc. 3rd ACM SIGCOMM Workshop Inf. Centric Netw.*, Hong Kong, 2013, pp. 35–36.

[58] A. Detti, C. Pisa, and N. B. Melazzi, "Modeling multipath forwarding strategies in information centric networks," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Hong Kong, 2015, pp. 324–329.

[59] S. Lederer, C. Timmerer, C. Westphal, and C. Mueller, "Adaptive video streaming over ICN," Internet Draft, draft-irtf-icnrg-videostreaming-03, IRTF, Reston, VA, USA, RFC 7933, Sep. 27, 2017.

[60] V. Jacobson *et al.*, "VoCCN: Voice-over content-centric networks," in *Proc. Workshop Re Architecting Internet*, Rome, Italy, 2009, pp. 1–6.

[61] L. Saino, I. Psaras, and G. Pavlou, "Hash-routing schemes for information centric networking," in *Proc. 3rd ACM SIGCOMM Workshop Inf. Centric Netw.*, Hong Kong, 2013, pp. 27–32.

[62] Z. Li and G. Simon, "Time-shifted TV in content centric networks: The case for cooperative in-network caching," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kyoto, Japan, 2011, pp. 1–6.

[63] 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.

[64] L. Ramaswamy and L. Liu, "An expiration age-based document placement scheme for cooperative Web caching," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 5, pp. 585–600, May 2004.

[65] E. J. Rosensweig and J. Kurose, "Breadcrumbs: Efficient, best-effort content location in cache networks," in *Proc. IEEE INFOCOM*, 2009, pp. 2631–2635.

[66] Y. Li, T. Lin, H. Tang, and P. Sun, "A chunk caching location and searching scheme in content centric networking," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Ottawa, ON, Canada, 2012, pp. 2655–2659.

[67] G. Zhang, B. Tang, X. Wang, and Y. Wu, "An optimal cache placement strategy based on content popularity in content centric network," *J. Inf. Comput. Sci.*, vol. 11, no. 8, pp. 2759–2769, 2014.

[68] S. Eum, K. Nakauchi, Y. Shoji, N. Nishinaga, and M. Murata, "CATT: Cache aware target identification for ICN," *IEEE Commun. Mag.*, vol. 50, no. 12, pp. 60–67, Dec. 2012.

[69] J. Sung, J.-K. K. Rhee, and S. Jung, "Lightweight caching strategy for wireless content delivery networks," *IEICE Commun. Exp.*, vol. 3, no. 4, pp. 150–155, 2014.

[70] J. Wang *et al.*, "An optimal cache management framework for information-centric networks with network coding," in *Proc. IFIP Netw. Conf.*, Trondheim, Norway, 2014, pp. 1–9.

[71] M. Bilal and S.-G. Kang, "Time aware least recent used (TLRU) cache management policy in ICN," in *Proc. 16th Int. Conf. Adv. Commun. Technol. (ICACT)*, 2014, pp. 528–532.

[72] K. Cho *et al.*, "WAVE: Popularity-based and collaborative in-network caching for content-oriented networks," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Orlando, FL, USA, 2012, pp. 316–321.

[73] C. Bernardini, T. Silverston, and O. Festor, "MPC: Popularity-based caching strategy for content centric networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Budapest, Hungary, 2013, pp. 3619–3623.

[74] M. D. Ong, M. Chen, T. Taleb, X. Wang, and V. C. M. Leung, "FGPC: Fine-grained popularity-based caching design for content centric networking," in *Proc. 17th ACM Int. Conf. Model. Anal. Simulat. Wireless Mobile Syst.*, Montreal, QC, Canada, 2014, pp. 295–302.

[75] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, and K. Drira, "Cache coherence in machine-to-machine information centric networks," in *Proc. IEEE 40th Conf. Local Comput. Netw. (LCN)*, 2015, pp. 430–433.

[76] Y. Song, H. Ma, and L. Liu, "TCCN: Tag-assisted content centric networking for Internet of Things," in *Proc. IEEE 16th Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, Boston, MA, USA, 2015, pp. 1–9.

[77] Z. Li *et al.*, "ICN based shared caching in future converged fixed and mobile network," in *Proc. IEEE 16th Int. Conf. High Perform. Switch. Routing (HPSR)*, Budapest, Hungary, 2015, pp. 1–6.

[78] O. Hahm, E. Baccelli, T. C. Schmidt, M. Wahlisch, and C. Adjih, "A named data network approach to energy efficiency in IoT," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Washington, DC, USA, 2016, pp. 1–6.

[79] M. Pathan, R. Buyya, and A. Vakali, "Content delivery networks: State of the art, insights, and imperatives," in *Content Delivery Networks*. Heidelberg, Germany: Springer, 2008, pp. 3–32.

[80] M. Hofmann and L. R. Beaumont, *Content Networking: Architecture, Protocols, and Practice*. San Francisco, CA, USA: Elsevier, 2005.

[81] M. Rabinovich, J. Chase, and S. Gadde, "Not all hits are created equal: Cooperative proxy caching over a wide-area network," *Comput. Netw. ISDN Syst.*, vol. 30, nos. 22–23, pp. 2253–2259, 1998.

[82] M. Rabinovich and O. Spatscheck, *Web Caching and Replication*. Boston, MA, USA: Addison-Wesley, 2002. [Online]. Available: http://www.amazon.com/Web-Caching-Replication-Michael-Rabinovich/dp/0201615703

[83] G. Pallis and A. Vakali, "Insight and perspectives for content delivery networks," *Commun. ACM*, vol. 49, no. 1, pp. 101–106, 2006.

[84] E. Nygren, R. K. Sitaraman, and J. Sun, "The akamai network: A platform for high-performance Internet applications," *ACM SIGOPS Oper. Syst. Rev.*, vol. 44, no. 3, pp. 2–19, 2010.

[85] (2007). *Akamai Technologies*. Accessed: Sep. 25, 2016. [Online]. Available: https://www.akamai.com/

[86] J. Dilley *et al.*, "Globally distributed content delivery," *IEEE Internet Comput.*, vol. 6, no. 5, pp. 50–58, Sep./Oct. 2002.

[87] J. Liang *et al.*, "When HTTPS meets CDN: A case of authentication in delegated service," in *Proc. IEEE Symp. Security Privacy (SP)*, San Jose, CA, USA, 2014, pp. 67–82.

[88] M. Ruiz, M. Germán, L. M. Contreras, and L. Velasco, "Big data-backed video distribution in the telecom cloud," *Comput. Commun.*, vol. 84, pp. 1–11, Jun. 2016.

[89] D. Kutscher *et al.* (2015). *ICN Research Challenges*. [Online]. Available: https://tools.ietf.org/html/draft-irtf-icnrg-challenges-03

[90] M. Dräxler and H. Karl, "Efficiency of on-path and off-path caching strategies in information centric networks," in *Proc. IEEE Int. Conf. Green Comput. Commun. (GreenCom)*, Besançon, France, 2012, pp. 581–587.

[91] X. Tang and S. T. Chanson, "Coordinated en-route Web caching," *IEEE Trans. Comput.*, vol. 51, no. 6, pp. 595–607, Jun. 2002.

[92] H. Shen and S. Xu, "Coordinated en-route Web caching in multiserver networks," *IEEE Trans. Comput.*, vol. 58, no. 5, pp. 605–619, May 2009.

[93] M. R. Korupolu and M. Dahlin, "Coordinated placement and replacement for large-scale distributed caches," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 6, pp. 1317–1329, Nov./Dec. 2002.

[94] L. Fan, P. Cao, J. Almeida, and A. Z. Broder, "Summary cache: A scalable wide-area Web cache sharing protocol," *IEEE/ACM Trans. Netw.*, vol. 8, no. 3, pp. 281–293, Jun. 2000.

[95] G. M. Voelker *et al.*, "Implementing cooperative prefetching and caching in a globally-managed memory system," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 26, no. 1, pp. 33–43, 1998.

[96] B.-G. Chun *et al.*, "Selfish caching in distributed systems: A game-theoretic analysis," in *Proc. 23rd Annu. ACM Symp. Principles Distrib. Comput.*, 2004, pp. 21–30.

[97] N. Laoutaris, O. Telelis, V. Zissimopoulos, and I. Stavrakakis, "Distributed selfish replication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 12, pp. 1401–1413, Dec. 2006.

[98] V. Pacifici and G. Dán, "Content-peering dynamics of autonomous caches in a content-centric network," in *Proc. IEEE INFOCOM*, Turin, Italy, 2013, pp. 1079–1087.

[99] V. Pacifici and G. Dán, "Selfish content replication on graphs," in *Proc. 23rd Int. Teletraffic Congr.*, San Francisco, CA, USA, 2011, pp. 119–126.

[100] G. Dan, "Cache-to-cache: Could ISPs cooperate to decrease peer-to-peer content distribution costs?" *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 9, pp. 1469–1482, Sep. 2011.

[101] G. Rossini and D. Rossi, "Coupling caching and forwarding: Benefits, analysis, and implementation," in *Proc. 1st Int. Conf. Inf. Centric Netw.*, Paris, France, 2014, pp. 127–136.

[102] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. K. Ramakrishnan, "Optimal content placement for a large-scale VoD system," in *Proc. 6th Int. Conf.*, 2010, p. 4.

[103] A. Dabirmoghaddam, M. Mirzazad-Barijoug, and J. J. Garcia-Luna-Aceves, "Understanding optimal caching and opportunistic caching at 'the edge' of information-centric networks," in *Proc. 1st Int. Conf. Inf. Centric Netw.*, Paris, France, 2014, pp. 47–56.

[104] J. Dai, Z. Hu, B. Li, J. Liu, and B. Li, "Collaborative hierarchical caching with dynamic request routing for massive content distribution," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, 2012, pp. 2444–2452.

[105] D. Corujo, R. L. Aguiar, I. Vidal, J. Garcia-Reinoso, and K. Pentikousis, "Research challenges towards a managed information-centric network of things," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Bologna, Italy, 2014, pp. 1–5.

[106] D. Karger *et al.*, "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide Web," in *Proc. 29th Annu. ACM Symp. Theory Comput.*, El Paso, TX, USA, 1997, pp. 654–663.

[107] G. Barish and K. Obraczke, "World wide Web caching: Trends and techniques," *IEEE Commun. Mag.*, vol. 38, no. 5, pp. 178–184, May 2000.

[108] A. Rowstron and P. Druschel, "Storage management and caching in past, a large-scale, persistent peer-to-peer storage utility," *ACM SIGOPS Oper. Syst. Rev.*, vol. 35, no. 5, pp. 188–201, 2001.

[109] C. Williamson, "On filter effects in Web caching hierarchies," *ACM Trans. Internet Technol.*, vol. 2, no. 1, pp. 47–77, 2002.

[110] C. Kumar and J. B. Norris, "A new approach for a proxy-level Web caching mechanism," *Decis. Support Syst.*, vol. 46, no. 1, pp. 52–60, 2008.

[111] B. Ager, F. Schneider, J. Kim, and A. Feldmann, "Revisiting cacheability in times of user generated content," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM)*, San Diego, CA, USA, 2010, pp. 1–6.

[112] L. Zhang *et al. Named Data Networking (NDN) Project*. Oct. 2010, accessed: May 10, 2017. [Online]. Available: https://named-data.net/wp-content/uploads/TR001ndn-proj.pdf

[113] H. Jeon, B. Lee, and H. Song, "On-path caching in information-centric networking," in *Proc. 15th Int. Conf. Adv. Commun. Technol. (ICACT)*, Pyeongchang, South Korea, 2013, pp. 264–267.

[114] G. Xylomenos, X. Vasilakos, C. Tsilopoulos, V. A. Siris, and G. C. Polyzos, "Caching and mobility support in a publish-subscribe Internet architecture," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 52–58, Jul. 2012.

[115] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie, "Optimal cache allocation for content-centric networking," in *Proc. 21st IEEE Int. Conf. Netw. Protocols (ICNP)*, Goettingen, Germany, 2013, pp. 1–10.

[116] H. Che, Y. Tung, and Z. Wang, "Hierarchical Web caching systems: Modeling, design and experimental results," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 7, pp. 1305–1314, Sep. 2002.

[117] S. Eum, K. Nakauchi, M. Murata, Y. Shoji, and N. Nishinaga, "CATT: Potential based routing with content caching for ICN," in *Proc. 2nd ACM SIGCOMM Workshop Inf. Centric Netw.*, Helsinki, Finland, 2012, pp. 49–54.

[118] B. Feng, H. Zhou, H. Zhang, J. Jiang, and S. Yu, "A popularity-based cache consistency mechanism for information-centric networking," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Washington, DC, USA, 2016, pp. 1–6.

[119] J. Li *et al.*, "Popularity-driven coordinated caching in named data networking," in *Proc. 8th ACM/IEEE Symp. Archit. Netw. Commun. Syst.*, Austin, TX, USA, 2012, pp. 15–26.

[120] Z. Ming, M. Xu, and D. Wang, "Age-based cooperative caching in information-centric networking," in *Proc. 23rd Int. Conf. Comput. Commun. Netw. (ICCCN)*, Shanghai, China, 2014, pp. 1–8.

[121] X. Hu and J. Gong, "Distributed in-network cooperative caching," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, vol. 2. Hangzhou, China, 2012, pp. 735–740.

[122] T. Wei, L. Chang, B. Yu, and J. Pan, "MPCS: A mobility/popularity-based caching strategy for information-centric networks," in *Proc. IEEE Conf. Glob. Commun. Conf. (GLOBECOM)*, Austin, TX, USA, 2014, pp. 4629–4634.

[123] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Modeling data transfer in content-centric networking," in *Proc. 23rd Int. Teletraffic Congr.*, San Francisco, CA, USA, 2011, pp. 111–118.

[124] I. Psaras, R. G. Clegg, R. Landa, W. K. Chai, and G. Pavlou, "Modelling and evaluation of CCN-caching trees," in *NETWORKING 2011*. Heidelberg, Germany: Springer, 2011, pp. 78–91.

[125] L. Muscariello, G. Carofiglio, and M. Gallo, "Bandwidth and storage sharing performance in information centric networking," in *Proc. ACM SIGCOMM Workshop Inf. Centric Netw.*, Toronto, ON, Canada, 2011, pp. 26–31.

[126] G. Carofiglio, V. Gehlen, and D. Perino, "Experimental evaluation of memory management in content-centric networking," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kyoto, Japan, 2011, pp. 1–6.

[127] S. K. Fayazbakhsh *et al.*, "Less pain, most of the gain: Incrementally deployable ICN," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 147–158, 2013.

[128] A. Gharaibeh, A. Khreishah, I. Khalil, and J. Wu, "Asymptotically-optimal incentive-based en-route caching scheme," in *Proc. IEEE 11th Int. Conf. IEEE Mobile Ad Hoc Sensor Syst. (MASS)*, Philadelphia, PA, USA, 2014, pp. 318–326.

[129] A. Gharaibeh, A. Khreishah, I. Khalil, and J. Wu, "Asymptotically-optimal incentive-based en-route caching scheme," *arXiv preprint arXiv*, vol. 1404, 2014. [Online]. Available: http://arxiv.org/pdf/1404.4639.pdf

[130] J. M. Wang, J. Zhang, and B. Bensaou, "Intra-AS cooperative caching for content-centric networks," in *Proc. 3rd ACM SIGCOMM Workshop Inf. Centric Netw.*, Hong Kong, 2013, pp. 61–66.

[131] M. Badov, A. Seetharam, J. Kurose, V. Firoiu, and S. Nanda, "Congestion-aware caching and search in information-centric networks," in *Proc. 1st Int. Conf. Inf. Centric Netw.*, Paris, France, 2014, pp. 37–46.

[132] W.-X. Liu, S.-Z. Yu, Y. Gao, and W.-T. Wu, "Caching efficiency of information-centric networking," *IET Netw.*, vol. 2, no. 2, pp. 53–62, Jun. 2013.

[133] T.-M. Pham *et al.* (2014). *Optimization of Content Caching in Content-Centric Network*. [Online]. Available: http://hal.upmc.fr/hal-01016470/file/caching_Paper_ICN_Final.pdf
[134] S. Wang *et al.*, "Could in-network caching benefit information-centric networking?" in *Proc. 7th Asian Internet Eng. Conf.*, Bangkok, Thailand, 2011, pp. 112–115.
[135] L. Zhang, J. Zhao, and Z. Shi, "LF: A caching strategy for named data mobile ad hoc networks," in *Proc. 4th Int. Conf. Comput. Eng. Netw.*, 2015, pp. 279–290.
[136] V. G. Vassilakis *et al.*, "A cache-aware routing scheme for information-centric networks," in *Proc. 9th Int. Symp. Commun. Syst. Netw. Digit. Signal Process. (CSNDSP)*, Manchester, U.K., 2014, pp. 721–726.
[137] J. J. Garcia-Luna-Aceves, "A fault-tolerant forwarding strategy for interest-based information centric networks," in *Proc. IEEE IFIP Netw. Conf. (IFIP Netw.)*, Toulouse, France, 2015, pp. 1–9.
[138] Y. Li, H. Xie, Y. Wen, C.-Y. Chow, and Z.-L. Zhang, "How much to coordinate? Optimizing in-network caching in content-centric networks," *IEEE Trans. Netw. Service Manag.*, vol. 12, no. 3, pp. 420–434, Sep. 2015.
[139] C. Bernardini. *SocialCCNSim*. Accessed: Feb. 15, 2016. [Online]. Available: https://github.com/mesarpe/socialccnsim
[140] I. Ud Din, S. Hassan, and A. Habbal, "SocialCCNSim: A simulator for caching strategies in information centric networking," *Adv. Sci. Lett.*, vol. 21, no. 11, pp. 3507–3511, 2015.
[141] D. Rossi and G. Rossini, *Caching Performance of Content Centric Networks Under Multi-Path Routing (and More)*, Relatório técnico, Telecom Paris Tech, Paris, France, 2011.
[142] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, "Impact of traffic mix on caching performance in a content-centric network," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Orlando, FL, USA, 2012, pp. 310–315.
[143] L. Saino, "On the design of efficient caching systems," Ph.D. dissertation, Dept. Electron. Elect. Eng., Univ. College London, London, U.K., 2015. [Online]. Available: http://discovery.ucl.ac.uk/1473436/1/thesis-final.pdf
[144] *Abilene Topology*. Accessed: Apr. 7, 2016. [Online]. Available: https://uit.stanford.edu/service/network/internet2/abilene
[145] A. Ghodsi *et al.*, "Information-centric networking: Seeing the forest for the trees," in *Proc. 10th ACM Workshop Hot Topics Netw.*, Cambridge, MA, USA, 2011, pp. 1–6.
[146] A. Wolman *et al.*, "On the scale and performance of cooperative Web proxy caching," *ACM SIGOPS Oper. Syst. Rev.*, vol. 33, no. 5, 1999, pp. 16–31.
[147] S. Ihm and V. S. Pai, "Towards understanding modern Web traffic," in *Proc. ACM SIGCOMM Conf. Internet Meas.*, Berlin, Germany, 2011, pp. 295–312.
[148] B. Feng, H. Zhou, S. Gao, and I. You, "An exploration of cache collaboration in information-centric network," *Int. J. Commun. Syst.*, vol. 27, no. 9, pp. 1243–1267, 2014.
[149] B. Feng, H. Zhou, and Q. Xu, "Mobility support in named data networking: A survey," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, no. 1, p. 220, 2016.

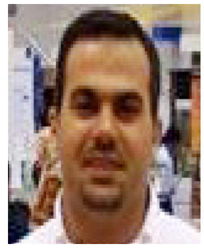
Ikram Ud Din received the M.Sc. degree in computer science and the M.S. degree in computer networking from the Department of Computer Science, University of Peshawar, Pakistan, in 2006 and 2011, respectively, and the Ph.D. degree in computer science from the School of Computing, Universiti Utara Malaysia (UUM), where he is a member of the InterNetWorks Research Laboratory. He also served as a IEEE UUM Student Branch Professional Chair. He has 10 years of teaching and research experience in different universities/organizations. His current research interests include resource management and traffic control in wired and wireless networks, cloud computing, traffic measurement and analysis for monitoring quality of service, mobility and cache management in information-centric networking, privacy preserving information lookup systems for information centric architectures, and access control mechanisms for distributed content storage.

Suhaidi Hassan (S'01–M'03–SM'08) received the bachelor's degree in computer science from the State University of New York in Binghamton, NY, USA, the master's degree in information science (telecommunication/networks) from the University of Pittsburgh, PA, USA, and the Ph.D. degree in computing (computer networks) from the University of Leeds, U.K. He is a Tenure Track Professor of computing network and the Founding Chair of the InterNetWorks Research Laboratory, School of Computing, Universiti Utara Malaysia (UUM). He has authored and co-authored over 250 refereed technical publications and successfully supervised 25 Ph.D. scholars in his research area of computer and communication networks. He was a recipient of the Swiss WKD Foundation's Young Scientist Fellowship Award at the World Knowledge Dialogue, Crans-Montana, Switzerland, in 2006. He is the Chair of the Internet Society Malaysia Chapter and the Internet Society Fellow alumnus to the Internet Engineering Task Force (IETF). He led a task force for the establishment of the International Telecommunication Union (ITU)-UUM Asia–Pacific Centre of Excellence for Rural ICT Development, a human resource development initiative of the ITU, which serves as the focal point for rural ICT development initiatives across the Asia–Pacific region. In addition to being a speaker at a number of renowned research conferences and technical meetings, he also participates in various international forums, such as ICANN meetings, Internet Governance Forums, the IETF, and the IEEE meetings. He has served as a Reviewer and a Referee for journals and conferences, as well as examiner for over 100 doctoral and postgraduate scholars in his research areas. He is also an IPv6 Auditor of the Malaysian Communication and Multimedia Commission, the Malaysian ICT regulator, auditing IPv6 implementation among Malaysian leading ISPs.

Muhammad Khurram Khan (M'07–SM'12) is a Full Professor with the Center of Excellence in Information Assurance, King Saud University, Saudi Arabia. He is an Adjunct Professor with the Fujian University of Technology, China, and an Honorary Professor with IIIRC, Shenzhen Graduate School, Harbin Institute of Technology, China. He has published over 325 research papers in the journals and conferences of international repute. He is an Inventor of 10 U.S./PCT patents. He has edited seven books/proceedings published by Springer-Verlag and IEEE. He was a recipient of the Outstanding Leadership Award at IEEE International Conference on Networks and Systems Security 2009, Australia, the Gold Medal for the Best Invention & Innovation Award at the 10th Malaysian Technology Expo 2011, Malaysia, and the Best Paper Award from the *Journal of Network & Computer Applications* (Elsevier) in 2015. He has been the Editor-in-Chief of the ISI-indexed international journal *Telecommunication Systems* (Springer-Verlag) since 1993 with an impact factor of 1.542 (JCR 2016), and the Founding Editor of *Bahria University Journal of Information & Communication Technology*. He is the full-time Editor/Associate Editor of several ISI-indexed international journals/magazines, including the IEEE COMMUNICATIONS SURVEYS & TUTORIALS, the *IEEE Communications Magazine*, the *Journal of Network & Computer Applications* (Elsevier), the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, IEEE ACCESS, *Security & Communication Networks*, the *IEEE Consumer Electronics Magazine*, the *Journal of Medical Systems* (Springer), *PLoS ONE*, *Computers & Electrical Engineering* (Elsevier), *IET Wireless Sensor Systems*, *Electronic Commerce Research* (Springer), the *Journal of Computing & Informatics*, the *Journal of Information Hiding and Multimedia Signal Processing*, and the *International Journal of Biometrics* (Inderscience). He is one of the organizing chairs of over five dozen international conferences and member of technical committees of over ten dozen international conferences. He is a fellow of the IET, U.K., the BCS, U.K., the FTRA, South Korea, a member of the IEEE Technical Committee on Security & Privacy, and a member of the IEEE cybersecurity community.

Mohsen Guizani (S'85–M'89–SM'99–F'09) received the bachelor's (with Distinction) and master's degrees in electrical engineering, and the master's and doctorate degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He is currently a Professor and the ECE Department Chair with the University of Idaho. He served as the Associate Vice President of Graduate Studies, Qatar University, the Chair of the Computer Science Department, Western Michigan University, and the Chair of the Computer Science Department, University of West Florida. He also served in academic positions with the University of Missouri–Kansas City, University of Colorado-Boulder, Syracuse University, and Kuwait University. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. He currently serves on the editorial boards of several international technical journals and is the founder and the Editor-in-Chief of *Wireless Communications and Mobile Computing* (Wiley). He has authored 9 books and over 400 publications in refereed journals and conferences. He has guest edited a number of special issues in IEEE journals and magazines. He also served as a member, chair, and general chair of a number of international conferences. He was selected as the Best Teaching Assistant for two consecutive years at Syracuse University. He was the recipient of the Best Research Award from three institutions. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker from 2003 to 2005.

Osman Ghazali received the Ph.D. degree in information technology (networking) from Awang Had Salleh Graduate School, Universiti Utara Malaysia. He is an Associate Professor and the Deputy Dean of the School of Computing, Universiti Utara Malaysia. He did his Post-Doctoral as a Research Scientist with the School of Engineering and Applied Science, Aston University in 2012. In 2011, he was the Head of Computer Science Department, School of Computing, Universiti Utara Malaysia, where he was the Technical Chairperson with the University Teaching and Learning Center from 2009 to 2011. He has over 100 publications as refereed book chapters and refereed technical papers in journals and conferences. He is a Senior Member of the InterNetWorks Research Laboratory. He is also a member of the ACM.



Adib Habbal is a Senior Lecturer of computing networks with the School of Computing, Universiti Utara Malaysia (UUM), and currently attached to the InterNetWorks Research Laboratory (IRL) as a IRL Seminar Coordinator. He also serves as IEEE UUM Student Branch Counselor and a Executive Council Member of the Internet Society Malaysia Chapter. He is actively involved in teaching, researching, and lecturing in the area of computer and communication networks. In 2013, he served the Internet Society (Geneva, CH/Reston, USA) as the ISOC Fellow to the 87th Internet Engineering Task Force (IETF) Meeting, Berlin, Germany. In 2013, he was selected as Asia–Pacific Advanced Network Fellow to the Techs in Paradise Conference, Hawaii. He was a recipient of the 2010 Excellent Service Award from the UUM for his outstanding achievements to the university in teaching and research. He has published many papers in reputed/refereed journals, presented several papers in national and international conferences, and successfully produced eight postgraduate scholars in the area of computer and communication networks. His professional activity includes being a speaker at a number of renown research conferences and technical meetings such as Internet2 and APNIC, a reviewer and referee for refereed journals and conferences on computing networks as well as an examiner for postgraduate scholars in his research areas. He is a professional member of the U.S.-based Institute of Electrical and Electronic Engineers, and actively involved in IEEE Communications Society.