A COMPREHENSIVE SURVEY OF INFORMATION-CENTRIC NETWORK: CONTENT CACHING STRATEGIES PERSPECTIVE

Mohammad Alkhazaleh[1], S. A. Aljunid[1] and Naseer Sabri[2]

[1]Advanced Communication Eng., Centre of Excellence, School of Computer and Communication Eng.
University Malaysia Perlis
[2]School of Computer Engineering, Al-Nahrain University, IRAQ

ABSTRACT

Information-Centric Networking (ICN) is an alternative to the current Internet infrastructure that relies on one-to-one communications. ICN aims to enhance internet performance based on the content name instead of IP addresses. ICN includes the in-network caching feature at every content router, which can reduce network traffic and enhance overall network performance efficiency. For that, many of caching strategies are proposed to manage and distribute the contents between the caches of content routers, which aim at improving numerous performance metrics such as cache-hit ratio, hop reduction ratio. This survey aims to provide a comprehensive review of the existing caching strategies that have not been mentioned in previous reviews and surveys as well as the old strategies mentioned. In this survey, the existing caching strategies are compared and analysed to clarify their purpose, decision-making, differences, advantages, and disadvantages. Also, the performance metrics, the simulation tools, and the network topologies are used to evaluate these caching strategies will be explained and illustrated. In the end, the issues related to the in-network cache will be mentioned and clarified in detail.

Keywords: Information-Centric Networking, Named Data Networking, In-network Caching, Caching Strategy, ICN Performance Metrics, ICN Evaluation Tools.

I. INTRODUCTION

Serving as a one-to-one conversation between two end hosts, the internet was developed decades ago to enable users to retrieve data from established servers. With the introduction of Transmission Control Protocol and Internet Protocol (TCP/IP) stack, users could transfer video, audio, and text packets across the internet via packet switching [1].

With the continuous and rapid growth of social networking, e-commerce, smartphone applications, and digital media, content distribution and retrieval become increasingly significant in internet communication. However, distribution issues arise when the internet is mainly used as a distribution network [2], such as when a lot of clients ask for the popular contents, at that time, the traffic explosion problem may occur. The process that applies a one-to-one communication protocol to solve distribution issues within the distribution network is complicated and likely to cause errors [2, 3]. Hence, ICN is introduced as a potential technique in designing the future internet.

ICN is an innovative standard that incorporates various architectures with a common notion: access to content and service granted by name instead of the original site or IP addresses [4]. In ICN, the in-network caching features are integrated at each content-router that has a significant influence on content retrieval. For example, the efficiency of ICN is high contingent upon how the caching resources are managed. Caching strategies that control what, when, and where content is kept within the network are used to properly manage caching resources [5]. Recently, the networking community has suggested several caching strategies to improve the performance of in-network caching in ICN Architectures. But these strategies differ in terms of principle, take a decision, and purpose. For that, this survey aims to provide a comprehensive review of the existing caching strategies that have not been mentioned in previous reviews and surveys as well as the old strategies mentioned. Moreover, these caching strategies will be explained in a simple and summarized manner with graphical examples to facilitate understanding of the working principle of the caching strategies. In this survey, the existing caching strategies will be compared and summarized to clarify their purpose, decision-making, differences, advantages, and disadvantages.

The existing caching strategies have been simulated with different tools, different performance metrics, and different network topologies. For that, performance metrics, simulation tools, and network topologies are used to evaluate these strategies will be explained and illustrated. At the end of this survey, issues related to In-network caching will be explained.

In this survey, section 2 provides a general overview of ICN architectures, especially, NDN. Review and comparison of the existing caching strategies are presented in Section 3. Sections 4 discuss performance metrics that are used to evaluate existing caching strategies in the ICN network. Evaluation tools and network topologies that are used to evaluate the existing caching strategies in ICN networks are presented in Section 5. Section 6 mentions some Issues related to the in-network caching of ICN.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

II. INFORMATION-CENTRIC NETWORKING ARCHITECTURES

With the initial introduction in TRIAD [6], that suggested information communication based-name, the ICN has then evolved and this leads to proposals of multiple architectures by researchers as shown in Figure-1. These architectures include a continuation of the Content-Centric Networking (CCN) [7] project; Data-Oriented Network Architecture (DONA) [8] from University California-Berkeley; the MobilityFirst [9] project, and the Named Data Networking (NDN) [10] project sponsored by the National Science Foundation (NCF); the Content Mediator architecture for content-aware nETworks (COMET) [11] project, the CONVERGENCE [12] project, the GreenICN [13] project, the Network of Information (NetInf) [14] project, a continuation of the Publish-Subscribe Internet Routing Paradigm (PSIRP) [15] project, the Publish-Subscribe Internet Technology (PURSUIT) [16] project, and the Scalable and Adaptive Internet Solutions (SAIL) [17] project sponsored by the EU Framework 7 Program (FP7). The GreenICN project is also backed by Japan. Although these approaches have different implementation methods, those architectures use the name to reach contents rather than the original site.

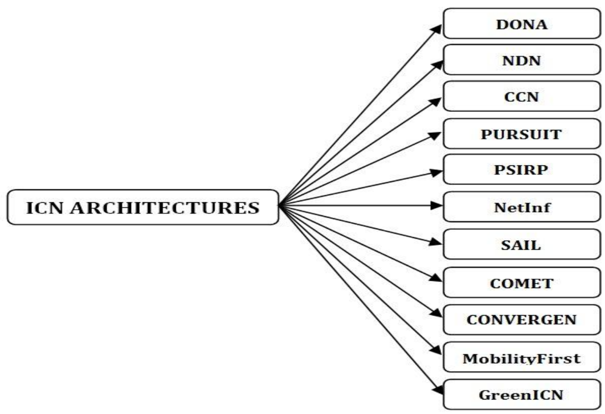


Figure-1. Information Centric Networking Architectures

NDN is considered the utmost prevalent architectures of ICN where a receiver-driven communication model is employed for the interaction of [1]–[3] content-routers by means of dual kinds of messages, namely the Interest and Data packets. Clients first direct the Interest-packets to retrieve contents and Data-packets would be generated with regards to Interest-packets. This one-to-one matching process would enable NDN to be flow-balanced. Figure 2 presents that the Interest-packet and Data-packet forwarding functions are performed via three-layered data-structures preserved in the NDN routers which are the Content Store (CS), a Pending-Interest Table (PIT), and a Forwarding-Information Base (FIB). The Content Store is a cache memory that stores promoted contents. The FIB is a list that compiles forwarding information with the known content prefixes, and the PIT list records the unfulfilled forwarded Interest-packets. Once a user directs an Interest-packet to an NDN router, a matching process is initiated with a search of the relevant content-name in the Content Store. Once a match is found, the producer's key would sign a Data-packet and forward the contents to the user. In the case of no match found, the content-name is examined in the PIT. The inward Interest-packet interface will be accumulated in the list of interfaces identified as Interest-aggregation when a PIT record match is successfully located. This would allow all concerned users to obtain a copy of the corresponding Data-packet when it is available. If the incoming Interest-packet is not found in the PIT, it is then channeled to the FIB of the router for the longest prefix match (LPM). If the FIB entry is matched by these LPMs, the Interest-packet will be delivered to the corresponding nearest router(s) and this is followed by the generation of a new PIT entrance with regards to the incoming interface. If the LPMs fail to match the Interest-packet, the router's forwarding policy can either drop the packet or flood all outgoing interfaces that have the same name of the initial Interest-packet with Interest NACK code that justifies the reason why the Interest-packet fails to be satisfied and forwarded. With the NACK code, appropriate measures can be implemented. Whenever the Data-packet is reverted to the NDN router, the matching process of the content name begins. It starts with a search in all PIT entries and if a matching PIT record is found, the entirely interfaces included in the table of incoming interfaces would receive the forwarded Data-packet. Then, the PIT entry is removed. According to the local caching strategy, the content will be kept in the CS to assist the requests of the same content in the future. The Data-packet will be dropped when a matching record does not exist (probably due to expired lifetime). Compared to IP, the support for multipath communications is simplified due to the associations between multiple faces and a single name.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

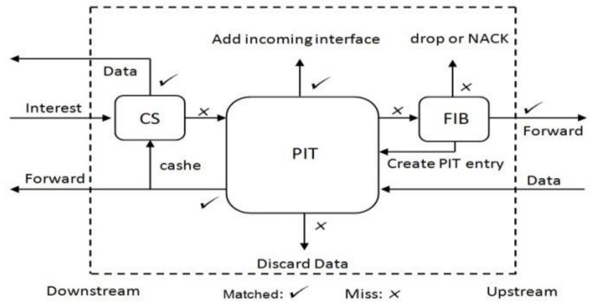


Figure-2. Forward processes in NDN.

III. IN-NETWORK CACHING

ICN includes in-network caching features at every content-router [5], which allow contents to be stored at some of the content-routers in the network, therefore provide efficient data delivery. The in-network caching considers one of the main advantages of the ICN, it allows users and content-routers to satisfy locally without going to servers, when consumer generate request for content, at any content-router must cache the content to support other users and content-routers maybe request the content and satisfy locally, this mechanism named placement, also which content to be replaced first when the cache is full, this mechanism named replacement.

This section presents an overview of existing caching strategies.

A. EXISTING CACHING STRATEGIES

This section explains the current caching strategies in a simple and summarized manner with graphical examples to facilitate understanding of the working principle of the caching strategies.

1) LEAVE COPY EVERYWHERE (LCE)

It is the default caching strategy in CCN / NDN, LCE leaves a copy of the content at each content router passing through it [18], as shown in Figure-3.

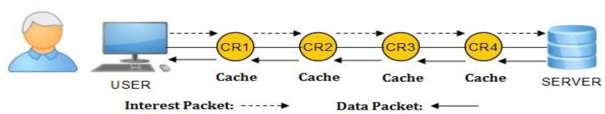
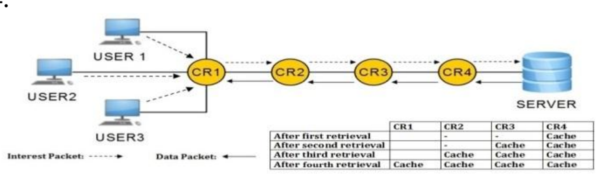


Figure-3. LCE operations

2) LEAVE COPY DOWN (LCD)

In the LCD [18], when the cache is hit, the LCD leaves a copy of the content on the neighbor content router towards the requester, as shown in Figure-4.

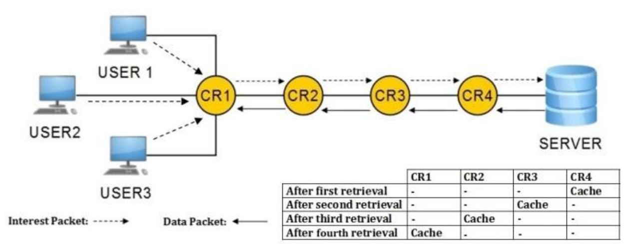


	CR1	CR2	CR3	CR4
After first retrieval		-	-	Cache
After second retrieval		-	Cache	Cache
After third retrieval		Cache	Cache	Cache
After fourth retrieval	Cache	Cache	Cache	Cache

Figure-4. LCD operations.

3) MOVE COPY DOWN (MCD)

In MCD [18], when the cache is hit, MCD moves the content copy on the neighbour content router towards the requester, as shown in Figure-5.



	CR1	CR2	CR3	CR4
After first retrieval	-	-	-	Cache
After second retrieval	-	-	Cache	-
After third retrieval	-	Cache	-	-
After fourth retrieval	Cache	-	-	-

Figure-5. MCD operations.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

4) CACHING WITH PROBABILITY (PROB(P))

When the content passes through the content router, the strategy generates a number between zero and one, if this number is less than the P, Prob(p) cache a copy of the content on the content router, otherwise pass the content to the neighbour content router without caching [18], as shown in Figure-6.

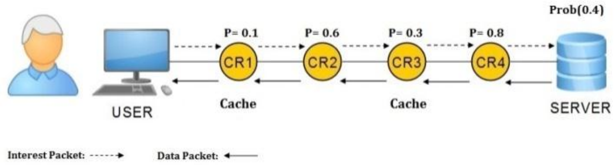


Figure-6. Prob(p) operations.

5) BREADCRUMB

It is similar to LCE, Breadcrumb leaves a copy of the content at each content router passing through it [19], as shown in Figure-7, but it is different because it stores additional information about the content in each content router, and this information helps to locate the content if evicted from the cache.

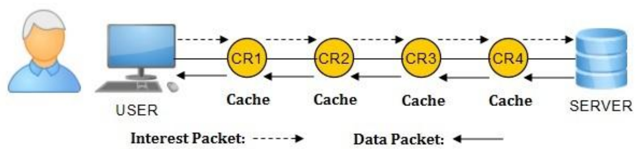


Figure-7. Breadcrumb operations.

6) PROBCACHE

It is a probabilistic caching strategy based on two factors; the first factor is TimeIn, which is the amount of traffic that it has to serve per unit time, the second factor is cache weight, which means the rate of the distance between user and content router and the distance between the user and server [20]. Probcache calculates the TimesIn and the cache weight, and then calculates caching probability (ProbCache(x)) at each content router, a router with a big (ProbCache(x)), compared to the content routers along the path, will be caching contents with higher probability, as shown in Figure-8.

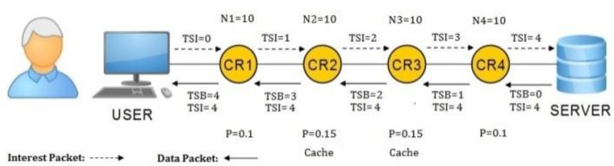


Figure-8. Probcache operations.

7) WAVE

It is similar to LCD but different as it divides the content into several chunks and each chunk needs an interest packet, after each request for this part, leaves a copy of the chunk at the neighbour content router towards the requester [21], as shown in Figure-9.

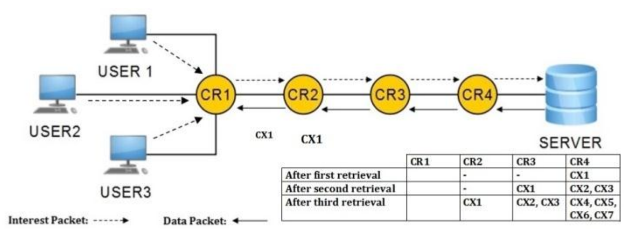


Figure-9. WAVE operations

8) A CHUNK CACHING LOCATION AND SEARCHING SCHEME (CLS)

It is similar to MCD, when the cache is hit, CLS moves the content copy to the neighbour content router towards the requester, but different as it moves the content copy at the neighbour content router towards the server when the copy of the content is evicted [22], as shown in Figure-10.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

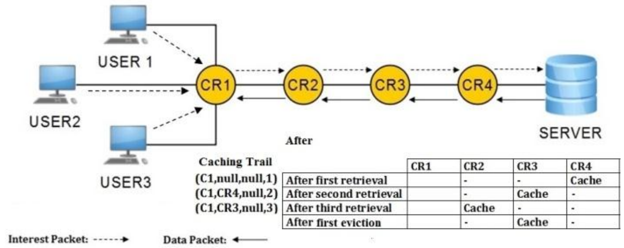


Figure-10. CLS operations.

9) EDGE

This strategy caches the content at the nearest content router for the requester along downloading path [23], as shown in Figure-11.

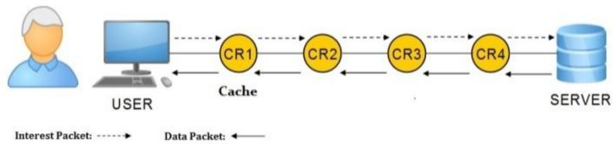


Figure-11. Edge operations

10) POPULARITY-BASED CACHING STRATEGY FOR CONTENT-CENTRIC NETWORKS (MPC)

The strategy caches only the popular content [24], in the beginning, the strategy counts the number of requests for the content at each content router along the path between the requester and the server, after that the content router which has more popularity than the threshold caches the content.

Given a cached popular content, the particular content router would send a message to its neighbouring content routers and recommend them to store the popular content. After that, the decision to cache the said content is made by the neighbouring content routers based on their respective caching policies as shown in Figure-12.

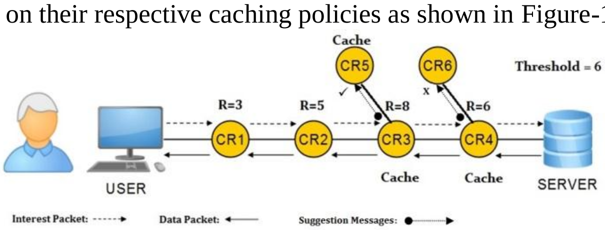


Figure-12. MPC operations

11) INTRA-AS CACHE COOPERATION

It is a collaborative caching strategy to prevent redundancy. It divides the network into a set of ASs and each AS has a set of content routers [25].

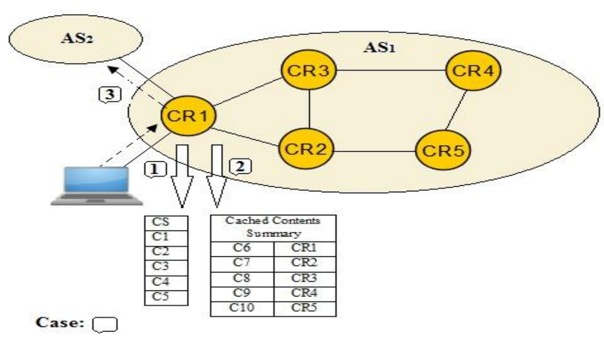


Figure-13. Intra-AS operations

Each content router inside the group summarizes what the contents contain (cached contents summary) and shares this summary with neighboring routers. If a request has been come for content at the content router, as shown in Figure-13, first, the strategy searches about the content in content router's Cache, if the content is Existing, it will send content to the requester, if the content is not Existing, it searches in the cached content summary, if the content is Existing, the interest message forwarded to the content router containing the content, otherwise the interest message is forwarded to another group.

12) CENTRALITY-BASED CACHING (CBC)

This strategy caches the content at most centralized content router between the requester and the server [26], as shown in Figure-14.

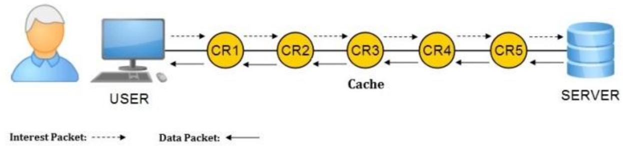


Figure-14. CBC operations

13) TWO LAYERS COOPERATIVE CACHING (TLCC)

In TLCC [27], the one autonomous system (AS) is divided into two groups as shown in Figure-15. The first group is the lower layer group (LLG), which connects with users and it uses to cache popular chunk. The second group is the upper layer group (ULG), which connects with other ASs or LLG and caches the chunk based on the key of hashed value. when a router of LLG group receive chunk, the router check if the hashed value within the range of the current router and the local popularity count is greater than the threshold, the strategy will cache the chunk in LLG router, otherwise, the chunk will send to the user without caching. When a router of ULG group receives chunk, the router checks if the hashed value within the range of the current router, if yes, the strategy will cache the chunk in ULG router. Otherwise, the strategy will forward the chunk to the dedicated router in the same group.

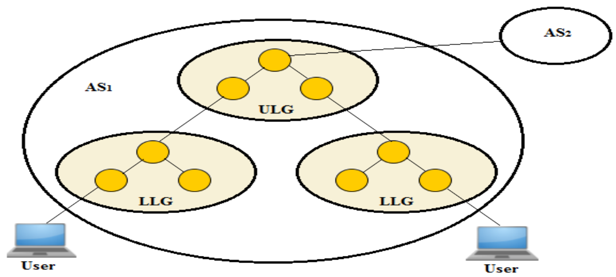


Figure-15. ULG and LLG inside AS

14) A DISTRIBUTED MAX-GAIN IN-NETWORK CACHING STRATEGY IN ICN (MAGIC)

The strategy calculates the local gain to cache the content at each content router that the interest packet passes through it, and the content router that has max gain caches the content [28] as shown in Figure-16. To calculate the local gain at each content router, magic calculates the place gain, and then calculate the cache replace penalty for all cached contents in the content router to git minimum cache replace penalty. After that, the local gain is calculated by subtracting the minimum cache replace penalty from the place gain.

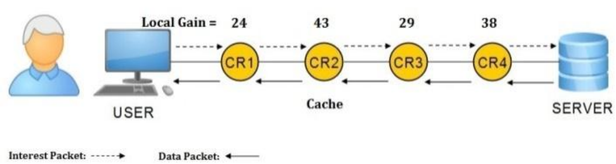


Figure-16. MAGIC operations.

15) *IN-NETWORK CASHING FOR ICN WITH PARTITIONING AND HASH-ROUTING (CPHR)* [29]

CPHR used the hash-routing and content space partitioning, for that, the content is partitioned into a number of pieces according to the hash value, and each piece is assigned to a distinct cache. In addition, the routers in each AS are classified into ingress router and egress router; the ingress router that could receive requests from their clients or other ASs, and egress router that connected to external links.

To implement CPHR, a new field is added in FIB named egress router that indicates to which egress router interest belonging, also new table is added at each ingress router named Partition Assignment Table (PAT), furthermore, two attributes are added in the interest packet named cache name and egress name, as shown in Figure-17.

When an Interest comes from an ingress link, the content router looks up its FIB to find the corresponding egress router and its PAT. Then, the interest packet is forwarded according to its name because the name of the assigned cache is prefixed to the name of interest. When the interest is received by the assigned cache, the strategy looks at the cache if it has corresponding content packet, if so, the strategy sends copy to the requester, otherwise, it replaces the name of the assigned cache of the interest's name with the value of egress name attribute of the interest and forward the interest.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

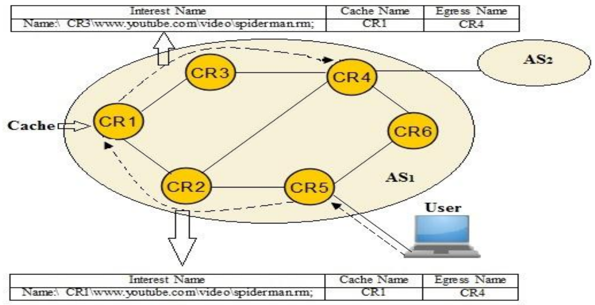


Figure-17. CPHR operations.

16) COOPERATIVE IN NETWORK CACHING (CINC)

CINC [30] is designed as a cooperative caching strategy for huge video streams such as live TV services. Using a set of deployed CRs, CINC aims to reduce the number of queries for time-shifted TV that is responded by servers beyond the ISP network. Figure-18 shows CINC that assigns sequence number for each content chunks and unique label for every content router to avoid redundancy, it caches the chunk at content router if the label of a content router equal to Z, which is illustrated in the following equation.

$$Z = Sd \bmod q \qquad (1)$$

Where q is number content routers, and Sd is the sequence number of a data packet. To achieve this objective, CINC includes two new tables in each content router, namely the Collaborative Router Table (CRT) and the Collaborative Content Store (CCS). CRT aims to document the information about the reachability of other content routers within a network, while CCS aims to document the content names cached by other content routers.

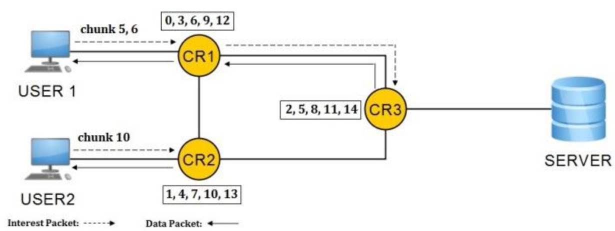


Figure-18. CINC operations.

17) AUCTION-BASED IN-NETWORK CACHING IN ICN (BIDCACHE)

In the BIDCACHE [31], when the request arrives at the content router, the content router decides to bid or not, if bid, its bid value will be bigger than the current value, if not, pass it to the next content router. The content router that has the highest bid wins the auction and caches a copy of the content, as shown in Figure-19. The bid is calculated based on a number of factors such as the size of the cache, the number of hops, delay, the value of LRU and other factors.

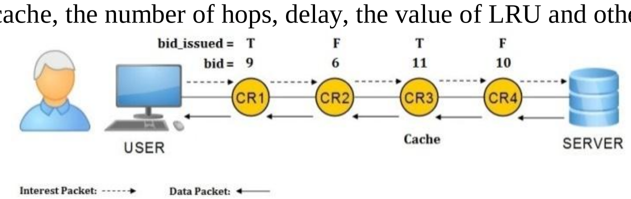


Figure-19. BidCache operations

18) LINK CONGESTION AND LIFETIME BASED IN-NETWORKING CACHING SCHEMES (LCLCS)

There are three factors that are used in LCLCS [32], link congestion, betweenness centrality, and content popularity. When the user requests content, the strategy calculates betweenness centrality at each content router along the interest packet path, the content router which has maximum betweenness centrality will cache the content, as shown in Figure-20. After the cache hit, the data packet is sent to the requester. When the data packet has arrived at the content router which has maximum betweenness centrality, the strategy calculates lifetime for content and records this value in the content router. Then if the cache has free space, the strategy caches content without eviction, if no, the content that has minimum lifetime is evicted to cache the new content. To calculate the lifetime, LCLCS calculates the delay saving as the time difference between obtaining the content from the original server and current content router, and then calculate betweenness centrality, in addition to the popularity weight and the cache time.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

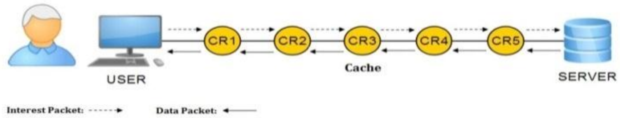


Figure-20. LCLCS operations

19) ADAPTIVE PRIORITIZED PROBABILISTIC CACHING ALGORITHM FOR CONTENT-CENTRIC NETWORKS (APP)

In APP [33], the data packet is divided into layers according to its importance, which is named data priorities. When the user request content, the request will be sent for all layers. The APP caches the content based on its priority because the priority is used to calculate the caching probability. When the content router receives data, it randomizes a real number between 0 and 1(0, 1), and this random number compares with caching probability. If the random number is less than or equal to the caching probability, the content will be cached at this content router as shown in Figure-21.

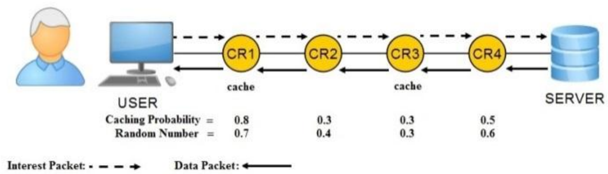


Figure-21. APP operations

20) POPULARITY AND GAIN-BASED CACHING SCHEME (PGBCS)

In PGBCS [34], when the interest arrives at the content router, the strategy checks if the content router has free space or not, then it calculates the caching gain. When the cache hit, if there are content routers that have free space along the downloading path, then PGBCS caches the chunk at the content router that has free space and nearest to the user, as shown in Figure-22. Otherwise, the strategy caches the content at the content router which has maximum caching gain, as shown in Figure-23. When the content arrives to cache at the content router, PGBCS check if the value of content chunk popularity is not less than the lowest value of content chunk popularity in the content router. After that the strategy deletes the content chunk which has the lowest value to cache the new content chunk, otherwise, the content chunk is discarded.

To calculate the caching gain, PGBCS calculates the average value of cached content chunks by using equation 2, and then calculates the utilization rate by using equation 3 and the distance factor by using equation 4. After that, it calculates the caching gain by using equation 5.

$$w_i = \frac{1}{n} \sum_{j=1}^{n} (O_j) \quad (2)$$

Where W_i is the average value of content chunks cached in the content router, O_j is content chunk j, and n is the number of content chunks cached.

$$p_i = \frac{S_i + r_i}{C_i} \quad (3)$$

Where S_i is the free space of content router, rj is the free space that expected to be occupied, and Ci is the cache size.

$$q_i = \frac{d_i}{D} \quad (4)$$

Where d_i is the number of hops between the user and the content router, and D is the number of hops between the user and the content source.

$$\mathrm{M(v_i)} = \frac{1}{\mathrm{w_i}} \cdot \frac{1}{\mathrm{p_i}} \cdot \frac{1}{\mathrm{q_i}} \quad (5)$$

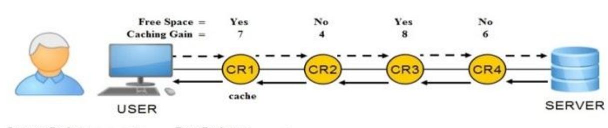


Figure-22. PGBCS operations (Case 1)

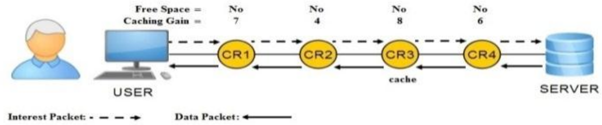


Figure-23. PGBCS operations (Case 2)

B. SUMMARY AND COMPARISON OF EXISTING CACHING STRATEGIES

The existing caching strategies differ in terms of principle, purpose, and the factors that are used to make a caching decision. So in this lesson, we will summarize the caching strategies in terms of the objective, the factors are used to make a caching decision, the advantages / benefits, and the disadvantages / challenges, to facilitate the comparison between the strategies, as shown in the table-1.

Table-1. Summary and comparison of exiting caching strategies

Caching Strategy	Author	Goal	Decision to cache	Advantages/Issues Tackled	Disadvantages/Challeng es
Leave Copy Everywhere (LCE), 2006	Nikolaos Laoutari s, etc.	Distribute content at content-routers	No	• Reduced onus • Quick content distribution • Simplicity	• Resource consumption • Cache frequency • No content distinction • Nonsingular position
Leave Copy Down (LCD), 2006	Nikolaos Laoutari s, etc	reducing the caching redundancy	Popularity + position	• Simplicity	• Resource consumption • Cache redundancy • No content distinction
Move Copy Down (MCD), 2006	Nikolaos Laoutari s, etc	reducing the caching redundancy	Popularity + position	• It improves cache efficiently by reducing caching redundancy. • Can be cached at router close to user after some requesting. • Simplicity. • position distinction	• No content distinction
caching with Probability (Prob(p)), 2006	Nikolaos Laoutari s, etc	Reducing the caching redundancy. Improving the efficiency.	Random	• It improves cache efficiently by reducing caching redundancy. • It improves cache hit ratio	• No position distinction
Breadcrumbs , 2009	Elisha J. Rosensw eig & Jim Kurose	best-effort caching, allow explicit coordination between caches.	No	• Making cached content available for off-path requests.	• Resource consumption • Cache redundancy • No content distinction. • No position distinction.
Probcache , 2011	IoannisP saras, etc	Permission of caching space for additional streams allocation (part of) the similar pathway, and justly multiplex contents of dissimilar streams amongst caches of a common	Distance between cached content-router and requesting content-router	• It improves caching powerfully by decreasing cache redundancies.	• Unfair allocation of capacity resources (unfulfilled goal) • Extra re computation of cache-request, cache-content reply packets • No content distinction

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

		pathway.			
WAVE , 2012	Kideok Cho, etc	Effectual content distribution and cache practice though dropping the overhead of cache managing	File-level Popularity + position	• Fast content dissemination	• Resource consumption • Cache redundancy • Update of content reply packets (computational overhead) • No content distinction • Applicable to object requests alone
A chunk-caching position and searching scheme (CLS), 2012	Yang Li, etc	CLS seeks to improve file download time and server workload.	Popularity + position	• It improves cache efficiently by reducing caching redundancy. • Making stored content obtainable for off-path needs. • Position distinction.	• No content distinction. • Definition of a threshold.
Popularity-based Caching Strategy (MPC), 2013	C´esarB ernardini , etc	the key focus of MPC is the distribution of popular content towards all neighboring content-routers (on-path & off Path)	Popularity	• Content distinction.	• Definition of a threshold
Edge Caching, 2013	Feixiong Zhang, etc	Decrease the sum of required hops to reach a content base.	Position	• Simplicity. • Reduction of hops traversed.	• Tendency to cache on specific content-routers. • No content distinction.
Intra-AS Cache Cooperation (Intra-AS), 2013	Jason Min Wang,	Dropping the content-cache redundancies through cached content visiblity to neighboring content-routers.	Cooperation , cache size of content-router, neighboring content-routers	• It improves cache efficiently by reducing caching redundancy. • Allowing off-path requests	• No content distinction. • No position distinction • Some time lead to access delay.
Centrality-based caching (CBC), 2013	Wei Koong Chai, etc	aims to: (1) reduce the latency , (2) reduce traffic and congestion (3) alleviate server load	Number of shortest paths	• Position distinction. • It improves cache efficiently by reducing caching redundancy.	• Tendency to cache closer to sources • No content distinction
Two layers cooperative caching (TLCC), 2014	Kyi Thar, etc	Using consistent hashing to eliminating duplicate contents.	Cooperation + Hash-Routing	• It eliminates the overlap contents.	• No content distinction. • No position distinction • Some time lead to access delay.
ADistributed MAx-Gain In-network Caching Strategy (	Jing Ren, etc	Reduce bandwidth consumption with a joint consideration of hop reduction and	Popularity + hops	• Content distinction. • Position distinction. • It reduces caching processes bases on I/O actions.	• Complexity, it calculates the replacement penalty for all cached contents in every

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

MAGIC), 2014		content popularity.		• It reduces the network bandwidth consumption • hop reduction • cache placement and replacement at the same time	content router along the interest path to content source.
In network Caching for ICN with Partitioning and Hash-Routing (CPHR), 2015	Sen Wang, etc	The objective of CPHR is to fully utilize ICN's built-in caching capability.	Cooperation + Hash-Routing	• Making cached content available for off-path requests. • It improves cache efficiently by reducing caching redundancy.	• No content distinction. • No position distinction • Some time lead to access delay.
Cooperative In-Network Caching (CINC), 2015	Zhe Li, Gwendal Simon	CINC aims to decrease the sum of enquiries for time shifted TV which are responded by servers beyond the ISP network.	Cooperation + number of routers	• It improves cache efficiently by reducing caching redundancy. • Increase of content diversity • Supporting the off-path caching.	• Designed for large video streams only. • No content distinction. • No position distinction • Some time lead to access delay.
Auction-based in-Network Caching (BidCache), 2016	Amritpal Singh Gill ,etc	Increase cache hit ratio and reduces the latency.	Bandwidth + LRU + LFU + Cache size + Hops + Link Delay	• Content distinction. • Position distinction. • It enhances the cache hit ratio of routing content-routers • It reduces users' requests time delay	• It is not easy to apply. • Complexity.
Link Congestion and lifetime Based In-Networking Caching Schemes (LCLCS), 2017.	Jiahui Kong, etc	LCLCS aims to sustain acceptable cache rate and significantly optimize network delay	Link congestion+ Betweennes s centrality + popularity.	• Content distinction. • Position distinction. • It enhances the cache hit ratio.	• Tendency to cache on specific content-routers. • It increases number of cache operations
Adaptive Prioritized Probabilistic Caching Algorithm (APP), 2017	Warit Sirichote dumrong , etc	aims to provide solutions for the limitations of previous caching algorithms	popularity + priority.	• Contents distinction. • It fulfills the applicant with the improved copied data superiority.	• No position distinction.
Popularity and gain based caching scheme (PGBCS), 2017	Zhandon g Fan, etc	It is design to obtain high gain.	Popularity + hops + content value	• Content distinction. • Position distinction. • It enhances the cache hit ratio of routing content-routers • It reduces network latency	• How to calculate the free space that expected to be occupied

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

IV. PERFORMANCE METRICS

This section presents performance metrics that are used to evaluate the existing caching strategies in ICN. The networks performance metrics can be divided into two categories, namely network and cache established criterions [35].

1) Network criterions: By evaluating the distribution of network among the content-routers within a network, the traffic rates within a net beside the quality of serviced practiced by clients, network-based metrics can determine the effectiveness of caching strategies. These metrics are estimated per network-scale or per forwarding path such as an AS or an ISP.

a) Hop-count: metric indicates the number of content-routers presence crossed by a content demand until it is fulfilled. Known as hop sum, decrease or count ratios, this metric is expressed in terms of fraction and is used to estimate the traffic decrease attained inside a net because of caching. The fraction is adjusted with the number of content-routers being crossed by a content demand when no caching. The hop count metric is proven to be a suitable estimator for deferral value within a net. When this metric increases, the traffic within a network increases.

b) Load fairness: metric refers to the ratio of the total number of content needs and content responses acknowledged by a content-router to the sum of content needs and content responses initiated within a network. This metric is employed to identify the presence of burdened content-routers.

c) Download time: metric represents the delay between the initiation of content demand and the coming of the consistent content. The key purposes of in-networking caching are to reduce delay and traffic. The transfer time is also known as delay time, Round-Trip Time (RTT) or latency time.

d) Network traffic: metric is represented by the total of link traffic values experienced within a network. It is widely employed to decide the traffic that exits an ISP. Therefore, ISP depends on its parent tiers or peers.

e) Link traffic: it shows the traffic volume on a connection. This metric is used to examine whether the traffic is concentrated to certain parts of a network or is being spreaded within a network.

f) Responses per request: metric represents the sum of content responses being gotten for a content demand. This criterion determines the network traffic caused by the multipath events.

2) Cache criterions: they are employed to evaluate the effectiveness of caching strategies via the assessment of their abilities to save and preserve the wanted content. These criterions are generally estimated per content-router.

a) Cache-content hit: it represents the sum of content demands fulfilled by the cache of a content-router, instead of the source content-router. It is represented by a fraction know as cache-hit percentage and is adjusted by the sum of content demands initiated within a network. This metric can be used to identify the load reserving of a server given caching is performed. Similar to server hit average, cache-content hit average is also a contentious criterion. An increase in this metric indicates better-caching policies. Sometimes, the metric is known as the cache-content hit probability or the cache-content hit ratio.

b) Server hits: metric represents the sum of content demands fulfilled by a server, hence, specify the respective server loads. It is represented by a fraction know as server-hit ratio and is adjusted by the sum of content demands initiated within a net. This metric can be used to identify the contents reserving of a server given caching is performed. A decrease in this metric indicates better-caching policies. Sometimes, the server hit average is also known as server-hit reduction ration.

c) Cache-content evictions: metric represents the sum of cache substitutes that happen on a content-router, based on a cache replacement strategy. Limited exploitation of cached content corresponds to an increase in this metric. Sometimes, this criterion is known as cache-content evictions rate or cache-content replacements rate.

d) Caching-content Frequency: this criterion is measured by a fraction of the sum of contents that are saved at a content-router to the sum of contents navigated over a similar content-router.

e) Caching-content Efficiency: it denotes to the sum of content demands fulfilled by a cache. It is measured in terms of a portion of the cache-hits on a content-router to the sum of contents cached in the cache of a similar content-router.

f) Cache-content redundancy: metric represents the sum of identical contents stored within a network. The cache diversity metric and cache redundancy metric are complementary in nature. An increase in cache diversity metric and a decrease in cache redundancy metric indicate good caching policies.

g) Cache-content diversity: it represents the sum of unique contents cached within a net. The metric depicts the trade-off between the content similarity metric and the rendering of caching strategies.

h) Absorption-content time: metric indicates the duration that content persists saved in the cache of a content-router. The cache replacement strategy and the popularity distribution of the content will determine the absorption time. Sometimes, the criterion is also known as caching time.

i) Stretch ratio: metric refers to the proportion of path being traveled to retrieve content. The total hops starting at the requester to the content-router that caches the content and total hops from requester till the base server.

ISSN- 2394-5125 VOL 7, ISSUE 04, 2020

j) Average number of caching processes: metric refers to the sum of I/O actions of cache memory/storing at all content-routers in average.

V. EVALUATION TOOL AND NETWORK TOPOLOGIES

This section presents evaluation tools and network topologies that are used to evaluate the existing caching strategies in ICN networks.

A. EVALUATION TOOLS

There is a wide variety of open-source evaluation tools are used to simulate different ICN architectures. These tools are available in multiple programming languages, and often customized for specific operating systems as shown in Table-2. But when we look at this table, about half of the tools are for CCN and NDN.

Table-2. Summary of Evaluation Tools

	Software	Project	Operating system	Language
OTHERS	Blackadder	PURSUIT	Linux, FreeBSD	C++,C, Python, Ruby, Java
	ICNsim	PURSUIT	Linux	C++,C
	openNetInf	SAIL	Portable	Java
	NetInf (nilib)	SAIL	Linux	C
	GIN	SAIL	FreeBSD	C(PHP)
	PeerKit	Convergence	Portable	Java
	Conet	Convergence	Linux	C
	XIA	XIA	Linux, MacOS	C (Python)
	MobilityFirst	MobilityFirst	Linux, Android	C++
	Icarus	GreenICN	Linux, MacOS	Python
CCN/NDN	CCNx	CCNx	Linux	C (Java)
	NFD	NDN	Linux, MacOS, FreeBSD	C++
	Mini-CCNx	CCN/NDN	Linux	C, Python
	CCN-Joker	CCN/NDN	Linux, Android	Java
	CCN-Lite	CCN/NDN	Linux, Android, MacOS	C
	Ns3-DCE CCNx	CCN/NDN	Linux	C(Python)
	CCNPL-Sim	CCN/NDN	Linux	C++
	NDNsim	NDN	Linux, MacOS	C++(Python)
	ccnSim	CCN/NDN	Linux	C++

The factors that affect the performance of in-network caching when simulating caching strategies:

1) Network Topology, The performance of the strategy differs when applied in different Topologies, so the simulations must be performed on more than one topology.

2) Cache Size, the small cache size increases the number of caching operations(placement and replacement) and reduce the cache hit ratio, also the large cache size maybe cause user delay when searching for specific content in a large cache.

3) Routing protocols, several routing protocols are designed for ICN architectures. For that, what the routing protocol is more appropriate for the strategy. Such as, the magic strategy uses OSPF protocol to get the sum of hops from the content-router to the original server.

4) The number of contents, the catalog size of cashed contents in a network simulation highly affects the performance. A large number of contents give more accurate results than the small number.

5) Content size: the size of the content effects at the performance. Such as, when the content size is large, few contents will be cached in the cache of the content router, which reduces the cache hit ratio.

6) Popularity distribution (Zipf), the number of requests for content at all content-routers, increase of this factor leads to generate traffic at the network.

B. NETWORK TOPOLOGIES

Many topologies are used to simulate the existing caching strategies; these topologies are classified into realistic topologies and Synthetic topologies.

Realistic topologies: real ISP topologies taken from Rocketfuel [23], [30], [36], [25], [29] and CAIDA AS-level trace [26] , and The Internet Topology Zoo dataset [37].

Synthetic topologies: generate from difference tools and models, such as GT-ITM[21], BRITE [25], [38], Watts-Strogatz (WS) model [39], and Barabasi-Albert model [26], [40], [31]

Some simulators like a SocialCCNSim contain some of these topologies such as Abilene, GEANT, DTELECOM, Tiger, and every networkx graph can be used as topology, by placing it on the graphs folder.

VI. ISSUES RELATED TO ICN CACHING

ICN has witnessed a massive interest by researchers and developers since the optimum performance is still a crucial target to achieve. Specifically, caching techniques of ICN gain much improvement with different strategies that tackle to efficiently enhance the caching of ICN.

Many challenges were introduced in caching technique of ICN, where the future enhancements are still required, below are a brief of the most ICN caching technique issues:

1) Cache placement: when consumer generates a content request, at any content-router must cache the content to support other users, and content-routers maybe request the content and satisfy locally.

2) Cache replacement: which the content to replace first when the cache is full and new content is arrived to cache.

3) Content selection: how to select distinct content to cache at the content-router, such as popular content.

4) Position selection: how to cache the content at a distinct position.

5) Place of cache: where to place the cache, at all content-routers or in special content-routers, such as edge content-router.

6) Cache size: What cache size is allowable, homogeneous or heterogeneous?

7) Caching strategy effectiveness: how to enhance the performance of the caching strategy, such as increasing cache-hit ratio and reducing user delay and redundancy and bandwidth consumption and traffic.

CONCLUSION

In general, it’s clearly shown that caching strategies of future ICN technology need more enhancements to stand with the emerged challenges and optimizing of the ICN performance. ICN is a substitute to the existing Internet infrastructure, which includes in-network caching features at every content router since it provides the ICN with efficient performance and reliability. This survey presents a comprehensive overview of the most recent caching strategies that have not been mentioned in previous reviews and surveys as well as the old strategies mentioned. In more detail, the caching strategies are explained in a simple and summarized manner with graphical examples to facilitate understanding of the working principle of the caching strategies. Moreover, the existing caching strategies are compared and summarized to clarify their purpose, decision-making, differences, advantages, and disadvantages. The performance metrics also have presented beside the network topologies and evaluation tools. The in-network caching of the ICN is affected by many factors that disturb its performance when simulating caching strategies, those factors are described and discussed in detail. The main future research trends and issues of content caching strategies are summarized and expounded.

REFERENCES

[1] D. Saxena and I. I. T. Roorkee, “Named Data Networking: A Survey,” *Comput. Sci. Rev. Elsevier*, vol. 19, pp. 15--55, 2016.

[2] M. M. S. Soniya and K. Kumar, “A survey on named data networking,” *2015 2nd Int. Conf. Electron. Commun. Syst.*, no. Icecs, pp. 1515–1519, 2015.

[3] A. Afanasyev *et al.*, “Named Data Networking,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–

73, 2014.
[4] M. A. Yaqub, S. H. Ahmed, Á. D. Á. Ccn, and Á. N. D. N. Á. Pub, “Information-Centric Networks (ICN).” pp. 19–34.
[5] C. Bernardini, T. Silverston, and A. Vasilakos, “Caching Strategies for Information Centric Networking : Opportunities and Challenges,” pp. 1–14, Jun. 2016.
[6] D. R. Cheriton and M. Gritter, “TRIAD: A new next-generation Internet architecture,” *Comput. Sci. Dep. Stanford Univ.*, no. April, 2001.
[7] “Cicn.” [Online]. Available: https://wiki.fd.io/view/Cicn. [Accessed: 29-Jan-2018].
[8] T. Koponen *et al.*, “A data-oriented (and beyond) network architecture,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, p. 181, 2007.
[9] “NSF Mobility First project.” [Online]. Available: http://mobilityfirst.winlab.rutgers.edu/TechApproach.html. [Accessed: 29-Jan-2018].
[10] “NSF Named Data Networking project.” [Online]. Available: https://named-data.net/. [Accessed: 28-Jan-2018].
[11] “FP7 COMET project.” [Online]. Available: http://www.comet-project.org/. [Accessed: 29-Jan-2018].
[12] “FP7 CONVERGENCE project.” [Online]. Available: http://www.ict-convergence.eu/. [Accessed: 29-Jan-2018].
[13] “FP7 Green ICN project.” [Online]. Available: http://www.greenicn.org/. [Accessed: 29-Jan-2018].
[14] “FP7 NetInf project.” [Online]. Available: http://www.netinf.org/. [Accessed: 29-Jan-2018].
[15] “FP7 PSIRP project.” [Online]. Available: http://www.psirp.org/. [Accessed: 29-Jan-2018].
[16] “FP7 PURSUIT project.” [Online]. Available: http://www.fp7-pursuit.eu/PursuitWeb/. [Accessed: 29-Jan-2018].
[17] “FP7 SAIL project.” [Online]. Available: http://www.sail-project.eu/. [Accessed: 29-Jan-2018].
[18] 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.
[19] E. J. Rosensweig and J. Kurose, “Breadcrumbs: Efficient, best-effort content location in cache networks,” *Proc. - IEEE INFOCOM*, pp. 2631–2635, 2009.
[20] I. Psaras, W. K. Chai, and G. Pavlou, “Probabilistic in-network caching for information-centric networks,” *ICN’12 - ACM Proc. Information-Centric Netw. Work.*, pp. 55–60, 2012.
[21] K. Cho, M. Lee, K. Park, T. T. Kwon, Y. Choi, and S. Pack, “WAVE: Popularity-based and collaborative in-network caching for content-oriented networks,” *Proc. - IEEE INFOCOM*, pp. 316–321, 2012.
[22] Y. Li, T. Lin, H. Tang, and P. Sun, “A chunk caching location and searching scheme in Content Centric Networking,” *IEEE Int. Conf. Commun.*, pp. 2655–2659, 2012.
[23] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, and A. Ghodsi, “Less Pain , Most of the Gain : Incrementally Deployable ICN,” pp. 147–158, 2013.
[24] C. Bernardini, T. Silverston, and O. Festor, “MPC : Popularity-based Caching Strategy for Content Centric Networks,” pp. 3619–3623, 2013.
[25] J. M. Wang, J. Zhang, and B. Bensaou, “Intra-AS cooperative caching for content-centric networks,” *Proc. 3rd ACM SIGCOMM Work. Information-centric Netw. - ICN ’13*, p. 61, 2013.
[26] 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.
[27] K. Thar, S. Ullah, and C. S. Hong, “Two layers cooperative caching in Content Centric Networks,” 2014.
[28] J. Ren *et al.*, “MAGIC: A distributed MAx-Gain In-network Caching strategy in information-centric networks,” *Proc. - IEEE INFOCOM*, pp. 470–475, 2014.
[29] S. Wang, J. Bi, J. Wu, and A. V. Vasilakos, “CPHR: In-network caching for Information-Centric Networking with Partitioning and Hash-Routing,” *IEEE/ACM Trans. Netw.*, vol. PP, no. 99, pp. 2742–2755, 2015.
[30] Z. Li, G. Simon, Z. Li, and G. Simon, “for cooperative in-network caching Time-Shifted TV in Content Centric Networks : the Case for Cooperative In-Network Caching,” 2011.
[31] A. S. Gill, L. D’Acunto, K. Trichias, and R. Van Brandenburg, “BidCache: Auction-based in-network caching in ICN,” *2016 IEEE Globecom Work. GC Wkshps 2016 - Proc.*, pp. 0–5, 2016.
[32] J. Kong, L. Rui, H. Huang, and X. Wang, “Link congestion and lifetime based in-network caching scheme in Information Centric Networking,” *IEEE CITS 2017 - 2017 Int. Conf. Comput. Inf. Telecommun. Syst.*, pp. 73–77, 2017.
[33] W. Sirichotedumrong, W. Kumwilaisak, S. Tarnoi, and N. Thatphitthukkul, “Adaptive prioritized probabilistic caching algorithm for content centric networks,” *Eng. J.*, vol. 21, no. 6 Special Issue, pp. 11–22, 2017.
[34] Z. Fan, Q. Wu, M. Zhang, and R. Zheng, “Popularity and gain based caching scheme for information-centric networks,” vol. 7, no. 30, 2017.
[35] A. Ioannou and S. Weber, “A Survey of Caching Policies and Forwarding Mechanisms in Information-Centric Networking,” *IEEE Commun. Surv. Tutorials*, vol. 18, no. 4, pp. 2847–2886, 2016.
[36] D. Rossi and G. Rossini, “On sizing CCN content stores by exploiting topological information,” *Proc. - IEEE INFOCOM*, pp. 280–285, 2012.
[37] V. Sourlas, L. Gkatzikis, P. Flegkas, and L. Tassiulas, “Distributed cache management in information-centric networks,” *IEEE Trans. Netw. Serv. Manag.*, vol. 10, no. 3, pp. 286–299, 2013.
[38] C. Bernardini, T. Silverston, and O. Festor, “A comparison of caching strategies for content centric networking,”

2015 IEEE Glob. Commun. Conf. GLOBECOM 2015, 2015.
[39] S. Shailendra, S. Sengottuvelan, H. K. Rath, B. Panigrahi, and A. Simha, "Performance Evaluation of Caching Policies in NDN - an ICN Architecture," *2016 IEEE Reg. 10 Conf.*, pp. 1117–1121, 2016.
[40] I. Psaras, W. K. Chai, G. Pavlou, and S. Member, "In-Network Cache Management and Resource Allocation for Information-Centric Networks," vol. X, no. X, pp. 1–11, 2013.
[41] C. Yi, J. Abraham, A. Afanasyev, and L. Wang, "On the Role of Routing in Named Data Networking," 2014.
[42] L. Wang and A. K. M. M. Hoque, "OSPFN : An OSPF Based Routing Protocol for Named Data Networking," pp. 1–15, 2012.
[43] H. Dai, J. Lu, Y. Wang, and B. Liu, "A Two-layer Intra-domain Routing Scheme for Named Data Networking," pp. 2815–2820, 2012.
[44] A. Alyyan, "NLSR : Named-data Link State Routing Protocol," pp. 2–7, 2016.