International Journal of Electrical and Computer Engineering (IJECE)
Vol. 10, No. 2, April 2020, pp. 1438~1453
ISSN: 2088-8708, DOI: 10.11591/ijece.v10i2.pp1438-1453

Study and analysis of mobility, security, and caching issues in CCN

Rao Naveed Bin Rais[1], Osman Khalid[2]
[1]College of Engineering and Information Technology Ajman University, Ajman, United Arab Emirates
[2]Department of Computer Science COMSATS University Islamabad, Abbottabad Campus, Pakistan

Article Info

Article history:

Received Mar 29, 2019
Revised Oct 11, 2019
Accepted Oct 21, 2019

Keywords:

Caching
CCN
Handoff
Mobility
Networking
Security

ABSTRACT

Existing architecture of Internet is IP-centric, having capability to cope with the needs of the Internet users. Due to the recent advancements and emerging technologies, a need to have ubiquitous connectivity has become the primary focus. Increasing demands for location-independent content raised the requirement of a new architecture and hence it became a research challenge. Content Centric Networking (CCN) paradigm emerges as an alternative to IP-centric model and is based on name-based forwarding and in-network data caching. It is likely to address certain challenges that have not been solved by IP-based protocols in wireless networks. Three important factors that require significant research related to CCN are mobility, security, and caching. While a number of studies have been conducted on CCN and its proposed technologies, none of the studies target all three significant research directions in a single article, to the best of our knowledge. This paper is an attempt to discuss the three factors together within context of each other. In this paper, we discuss and analyze basics of CCN principles with distributed properties of caching, mobility, and secure access control. Different comparisons are made to examine the strengths and weaknesses of each aforementioned aspect in detail. The final discussion aims to identify the open research challenges and some future trends for CCN deployment on a large scale.

Corresponding Author:

Rao Naveed Bin Rais,
College of Engineering and Information Technology Ajman University,
Ajman, United Arab Emirates.
Email : [1]r.rais@ajman.ac.ae, [2]osman@cuiatd.edu.pk

1. INTRODUCTION

Mobile devices, such as personal devices, laptops, smartphones, and tablets having sensing capabilities and connected via wireless Internet have become popular worldwide [1]. Wireless Networking has surged in attractiveness and has contributed to the increase of the user desire to have ubiquitous connectivity. Resource-constrained devices such as sensors and smart objects are connected on a large scale. Besides, with the growing number of mobile devices, huge proportion of mobile networks are now part of the global networking system. Global mobile data traffic is expected to be increased 52-fold from 2013 to 2022 to reach around 77 Exabyte per month, and only in 2017, mobile video traffic accounts for 59 percent of total mobile traffic [2]. Moreover, a large number of devices involved in wireless networking are mobile in nature. This mobility of devices tends to modify network topology with the passage of time. Traditional end-to-end Internet communication protocols have bottlenecks in coping with such mobility and dynamic network conditions. In addition, many content providing companies such as newspapers are migrating to provide digital services, as their content are available online. Hence, the majority usage of Internet is information-centric rather than host-centric.

The increasing demands of new applications has raised the architecture's requirements to be more flexible in terms of content sharing, mobility management, secured connectivity, caching, privacy, etc. However, large number of such key requirements still cannot be addressed properly. Keeping in view the above-cited issues, new architectures and paradigms are required to be devised, investigated, and eventually deployed on large scale. For this purpose, Content-Centric Networking (CCN) [3] has arisen as an emerging candidate for upcoming content-centric Internet architecture.

CCN has gained considerable popularity and has become comparatively demanding architecture of the future Internet [3]. CCN is an emerging technology in which content is accessed using its name instead of the location of the host, which possesses the content, thereby resulting in efficient bandwidth utilization and reducing communication cost. CCN communication principle suggests that users (consumers) are more interested in content (what) rather than knowing about the owner of the content (who) and the location of the content (where), as long as the integrity of the content is maintained. Usually in CCN, media servers and web portals are used to host contents. To retrieve a content, an end-to-end connection is required to be established. For that purpose, CCN is a self-organized approach that enables to push only the relevant data when desired.

CCN differs from traditional TCP/IP network in many aspects. CCN constitutes in-network caching, name-based routing, location-independent content support, inherent security, and default multicast support, to name a few. CCN is an extension of the Information Centric Network (ICN) research projects comprising four main entities: the producers, the consumers, the publishers, and the routers. The Forwarding Information Base (FIB), Pending Interest Table (PIT), and Content Store (CS) are elements of a router in a CCN as shown in Figure 1. FIB replaces the traditional routing tables, while PIT is a cache for the requests. CS is an optional component and acts as an extra cache where requests (known as Interest) might be stored. Manifests, Interest packets, and content objects are types of messages dealt within a CCN. In CCN, each content has an associated location-independent name.

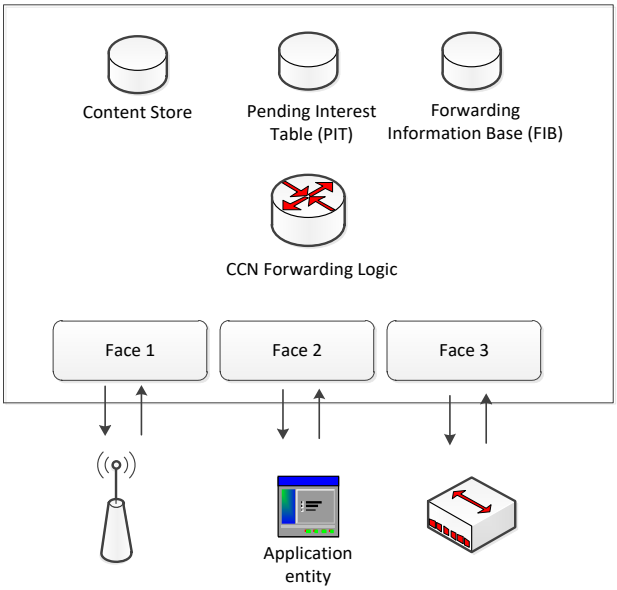

Figure 1. CCN Router Components

The receiver (consumer) of the content initiates CCN communication by sending an *Interest* packet. The content owner or holder may respond to an *Interest* packet by sending the *Data* packet. The data pieces in CCN are assumed self-identifying and self-authenticating unit, thereby providing an inherent security. While CCN promises to be a strong candidate for the network architecture of future due to its inherent flexibility and user-centric approach, there are a number of open research issues that need to be addressed before it can be used as an alternate to the status-quo Internet architecture. These research issues include naming, routing, caching, mobility management and security of CCN [4]. Although the research issues have been discussed by some studies (e.g., [1, 4]), yet these studies provide a basic overview of the issues without deeply investigating each of these. Moreover, there is no study, to the best of our knowledge, which targets an ensemble of these research issues at one single place. This paper is an attempt towards

filling the gap in the existing literature on CCNs. Specifically, we focus on three major areas of research in CCN, caching, security, and mobility management. The basic idea is based upon content management in terms of where to place content (caching), how to secure it from threats and maintain its integrity (security), and how does network mobility effects cached content and service to the consumer (mobility). Here, we provide an overview of technology, comprehensive information and a clear comparison of aforementioned three approaches. Moreover, we perform the quantitative comparisons of state-of-the-art selected caching schemes for different scenarios. The quantitative analysis provides an insight into how the caching performance effected with the variation in cache size and contents to be cached. The comparisons are performed using CCN-based simulator Icarus. The rest of the paper is organized as follows. In Section 2, we present mobility management schemes and issues related to CCN. Section 3 provides an analysis of CCN security schemes and relevant issues, whereas a detailed discussion on CCN caching studies is presented in Section 4. Finally, we conclude the paper in Section 5.

2. MOBILITY MANAGEMENT IN CCN

Current architecture of the Internet has been designed for fixed network environment where nodes require to have end-to-end communication for data transfer. In order to support mobile nodes over the network, existing Internet architecture requires several operations like multistage address resolution, frequent location updates, etc. IP networks are based on pre-established connection scheme where IP addresses of sender and receiver are used for the delivery of contents over a pre-established network. In such networks, frequent movement of mobile users may result in temporary or long-lived disconnection. Alternatively, in CCN, as content is accessed by its name instead of IP, an end-to-end connection is no longer required to retrieve content [3]. So a mobile user can achieve a seamless handoff while continuously accessing content at another location.

CCN mobility can be classified as a client (consumer) mobility or server-side (producer, publisher, or content router) mobility [5]. Client mobility deals with client movements during the time when the requests (*Interest*) for data objects is initiated. Likewise, server-side mobility refers to the handling of location changes made by object or set of objects. The server-side mobility might be because of the change in the content possession by another node in the network, or due to the mobility of the node while carrying the content. Moreover, a third type of mobility is network mobility in which a complete network changes location (e.g., a body area network or a vehicular network). In CCN, when a collection of complete object set is changing their location, the new path advertisement is needed to be broadcasted keeping the old routing records safe. This may also involve movements of some chunks of the objects belonging to a collection of objects to a new location (e.g., a company employee takes a laptop on a trip). Moving an entire (heterogeneous) network results in too many routing updates, which may result in bottlenecks unless they allow for relative route broadcast.

2.1. Classification of CCN content and mobility

Keeping in view the stability and mobility concerns in CCN, there are two types of contents named as data contents and user contents. *Data* content is significantly termed as access to the content itself. Persistent content providers are used to serve the data contents and dissemination of their replication for the ease of access to content. Therefore, locations e.g., Web pages, streaming services, and files, are comparatively stable. While user content, on the other hand, contains information having real-time characteristics requested before creation. Internet phone, chatting, and e-gaming are the family of user-content. Creation and management of the content is the responsibility of individual mobile devices. Contents' service components such as rendezvous points and control messages are used for serving data contents and user contents. A rendezvous point is basically a location manager used to keep track of locality information of all providers containing location information of peers. Moreover, a rendezvous point used as handler of mobility by point intervening in the relay of all *Interest* and *Data* packets. It is also known as indirection point. Control messages on the other hand are used to address the issues related to changing of content provider's routable prefixes. However, a sender-driven control message is needed for source mobility.

2.2. User mobility

Mobility for the buffered contents in CCN on user-side is supported by using re-transmission mechanism of *Interests* and *Data* packets, which first comes to ISP. When the service initiator performs a handoff, it receives a sequence number of the contents from its peer node. During communication, lost chunk of information data can be recovered by simple process known as re-transmission of data packets. To this end, Tsipoulus *et al.* [6] sectioned the CCN services as three types: real-time documents, channels, and on-demand

documents based on service characteristics, and present different forwarding techniques accordingly for each types of traffic. The study differentiates between Persistent *Interest* and the regular *Interest*. Persistent *Interest* is kept in PIT table for lifetime. Efficient streaming traffic is delivered having real-time and data loss-tolerant properties. In case of persistent *Interest*, after handoff, the route remains persistent to the old location and responses are multicast to the old location. This concept of multicasting may result in waste of network resources. Moreover, signaling message can be used if the frame router could not successfully deliver content corresponding to a persistent *Interest*, thereby increasing the failure ratio. When this value reaches a threshold, the edge router generates 'host unreachable' with the specific name that may be in the following form: **(persistent *Interest* name)/'host unreachable'**. This signaling-message is then delivered to the content server. Gateways used to communicate that signaling-message can abolish the related obsolete persistent request and responses from their PIT. When handoff occurs due to the mobile client, persistent *Interests* are retransmitted to a new path. Process of retransmission includes a dedicated bit that eliminates the outdated PIT entry. If the *Interest* is received on content server, it sets dedicated bit to 'on' in the next responses

2.3. Producer or router mobility

For receiving *Interests* smoothly after handoff, name of the mobile node is required to be updated in the routing table for each individual content. To this end, binding service between changing names is required at the both old and new locations. To change the content name, application server such as a rendezvous point or a proxy agent-binding server is useful to provide binding service [7]. Rendezvous-based approaches are used for changing names and binding services are discussed as follows:

a. Direct exchange for location update

In direct exchange, for location update when the service initiator handoff occurs, it indicates the data source mobility. The peer node is then informed about new classified pathname of the initiator for smooth running of ongoing service. Hence, the peer can continue getting replies even the handoff occurs. Recovery of lost data portion by simple re-transmission is impossible. In order to recover the lost data, application regenerates the *Interests* with the new hierarchical pathname. With the factors of simplicity and effectiveness it also causes issues when handoffs are performed simultaneously by the peers.

b. Query to the rendezvous for location update

On initiation of a service, the service initiator queries a rendezvous server for obtaining its peer's location record. As location information of all clients is maintained with rendezvous server, so on occurrence of a handoff, new path-name of the client is communicated to the rendezvous server. Therefore, using the rendezvous server, the peer can obtain the routable prefix even after simultaneous handoffs.

c. Mobility with indirection point

Another technique for reducing the handoff delay is the indirection scheme that can be considered. To initiate a request, the initiator issues a request and response packet with the name of its peer. On successful receiving of packet, it stores that packet and issues another *Interest* to the peer with its new routable name. On peer's response, the indirection server issues a corresponding response to the packet that is buffered and refer them towards the initiator. During an initiator handoff, it notifies occurrence of the handoff event towards the indirection server. However, on receiving *Interest* for the initiator, indirection server buffers that *Interest* and delays generating another corresponding *Interest* toward the respective node. When service initiator informs new hierarchical pathname, the indirection server continues to generate corresponding *Interests* and its service toward the initiator.

d. Interest forwarding

In order to cope with the drawbacks in previous schemes, layered approach known as *Interest* forwarding is proposed [7]. In this approach, new name is not required for seamless service location. Rather, it requires the modifiable routers as incoming *Interests* are buffered in CCN routers towards a handoff client. Overhead caused by addition of new entry routers on the way between the access routers to corresponding handoffs is reduced. In addition to this, handoff process does not require a new routable prefix, while handoff latency is diminished as well.

e. Proxy-based mobility management scheme [8]

This scheme is based on the concept that user proxies capable of supporting CCN are configured over traditional IP network. Contents are available to users via the proxy node when required. User sends a query packet to the proxy node instead of making connection with all other devices having required contents. CCN network, on receiving the query packet (*Interest*), tries to get content thus reducing the overhead of network configuration. The scheme is composed of following steps:

- Handover detection) By using the information of physical link or router advertisement mobile node detects the change network status.

- Handover indication) While the handover detection event is in process, a *hold* request message is sent to proxy node by the mobile node before detection of actual handover. On receiving the *hold* message, the proxy node stops the process of sending content data to MS's old location and only stores content data to its local repository for future re-transmissions.
- Handover complete) In case of acquiring new IP address, mobile node notifies the new IP to the proxy node by using handover message piggybacking the sequence of messages received on old location. The proxy node transmits the stored data to new location and does not store and transmit the repeated *Interest* packets. After receiving content packet on old location, mobile node can ask next content data segment using normal CCN *Interest* packet.

f. CSCM [9]

CSCM offers a dynamic strategy for caching in CCN for mobile nodes, while increasing the data availability, thereby improving network performance. The scheme considers a number of factors including mobility of nodes, content popularity and content diversity. As per CSCM functionality, a node while hosting a content and serving nearby nodes chooses a nearby node to handover the content before moving out of the location based on some criteria. It also keeps into consideration of dynamic weighting of each of the parameters used for making the decision for content storage.

g. Tunnel-Based Redirection (TBR)

This approach performs the redirection of incoming *Interest* packets between content router of Mobile Content Source's (MCS) home domain and the content router in the new domain by using tunneling [10]. To guarantee the presence of a router, it periodically broadcasts the prefix names. Point of attachment in CCN is located by the comparison of this broadcasted information, which is also used to know whenever a network initiates a prefix update round to the home domain. Home content router announces the prefix of the MCS name initially to inform the other Counter Router (CR). As per this scheme, the prefix update (PU) message about current movement is broadcasted by MCS and its active state is announced to the local routers of the domain. Then, the CR of the new domain sends the PU message to the home domain. The redirection path can be extracted by the exchange of the prefix update and its acknowledgement messages (PACK). The *Data* packets and *Interests* are then exchanged among the MCS and home CR through this name prefix. This scheme works in the following three steps as shown in Figure 2.

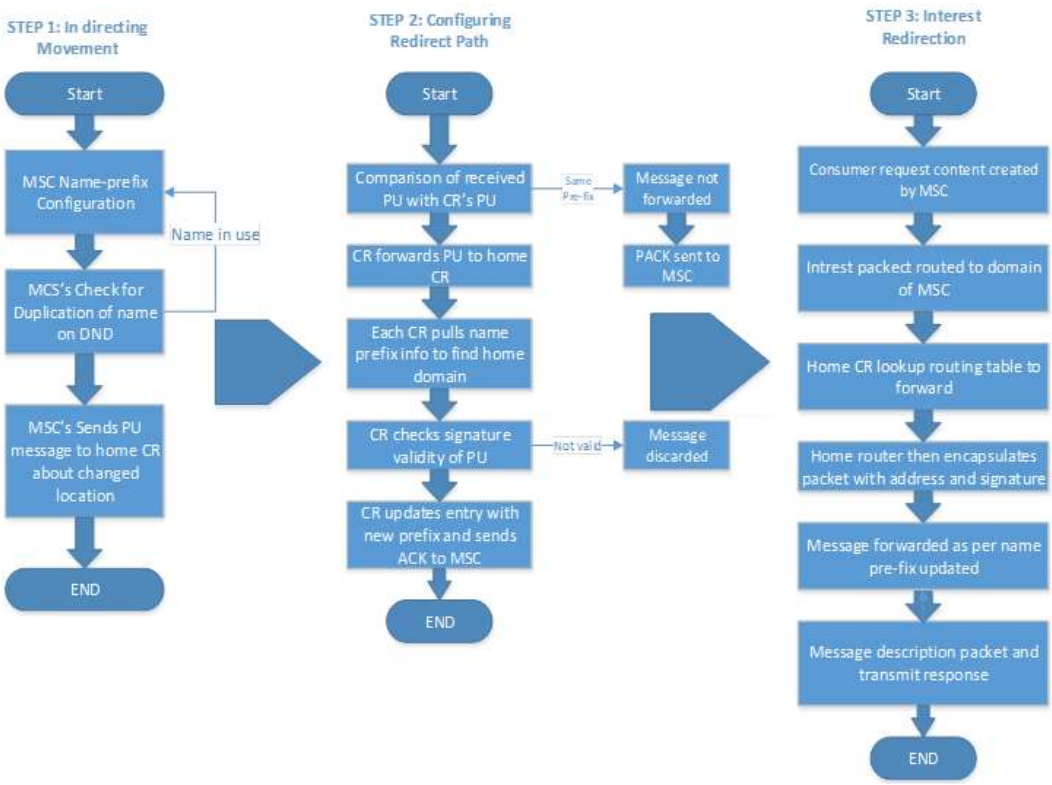

Figure 2. Tunnel based redirection (TBR) scheme

- Step 1: Indicating movement is the first step that identifies whether MCS would change the status of networks by the network address information or through the underlying physical link information. Normally, data is provided by a wireless AP to MCS containing all CRs in the access domain. When the change of domain occurs in network location, MCS forms a tentative name-prefix in order to tunnel the *Interest* packets, which comprises MCS's original name prefixes and new domain name. In order to check validity of tentative name prefix MCS performs duplicate name prefix detection (DND). *Interest* packets are broadcasted by MCS to the all in-range peripherals that are on a physical link residing in CR's domain that have been moved, for afore-cited reason. This broadcast checks whether its tentative name prefixes is used by someone else. If it is used by anyone else, then the MCS repeat the DND process for creating new tentative name. Eventually, MCS forwards the PU message indicating its new location to its home-based domain CR.
- Step 2: This step deals with the configuration of the redirection path. The CR that has received a PU message compares name prefix domain of the PU message with that of its own. CR sends the PU message to CR based on FIB reference. Otherwise, the PU message is not forwarded, instead a PACK message is sent to the MCS. When a PU message is received, each CR pulls out the name prefix information immediately from the router to find the message's home domain router, forming a PIT entry with the name-prefix. At that point, the home switch's entrance for that MCS is changed to the new one that the PU message has set. While getting the PACK message, middle of the road CRs look into the direction table once more.
- Step 3: This is the last step of the scheme and deals with Interest redirection. At last, the content consumer requests a content information made by the MCS, and intrigue correspondences are directed towards MCS. Home CR looks up in the routing table to send the Interest packet. The newly created Interest packet is encapsulated by the home router and forwarded to the tentative name prefix that was extracted from the routing table. Intermediate CRs between the home content router and the MCS deliver this message through regular CCN forward techniques. The MCS receives and decrypts the encapsulated Interest packet and transmit the Data response.

3. SECURITY IN CONTENT CENTRIC NETWORKS

In this section, we provide the security approaches in CCN and highlight the architectural weaknesses of generic CCN in terms of security [11, 12]. However, our analysis does not include implementation weaknesses due to faulty cryptography schemes and algorithm selection. The major issues discussed are the system inherent loopholes. CCNs are host independent and data dependent. Because of this reason, we need to associate the security protocols with either the naming schemes or the data rather than securing the channel or the machine. One of the major areas regarding research in security of CCNs is the naming convention [13]. Most of the currently proposed naming schemes provide integrity of content using public/private key pairs [13, 14]. The real-world objects are linked to their keys using PKI. Security risks relevant to naming schemes and PKIs still require more focus. Signatures are also included in packets to provide authentication [4, 15]. Certificate chains are hierarchical and require an inheritance-based system for its application. Moreover, in CCN, Name Data Networking (NDN) certificates follow name hierarchy and are easily implemented. The security of CCN is mainly linked to the naming schemes and architectures [16]. For human-readable names to be implemented, we require to trust a third-party for the verification of data and name. The third-party can be replaced if the network develops a trusted relationship with the system performing name resolutions. Although flat names support self-certification, which is extremely useful, they are not human-readable. In this situation, the third-party becomes mandatory and needs to be trusted since we require flat names to be mapped on by the human-readable names. When users reveal their interests in particular information that is to be transferred to it, the name of the information being requested is available to all the CCN nodes processing the given request [17].

Name resolution in CCN is performed in two stages. In the first stage, name of the content is resolved to one or more locators hosting the content. These locators can constitute over a set of nodes hosting the content altogether. In the second stage, the *Interest* packet is routed to a locator using any routing scheme in use. The *Interest* packet is forwarded based on the name of the content using the principle of name-based routing. While forwarding, some state information is maintained on the forwarders of the *Interest* so that the *Data* packet of the content follows the same path on its way to the consumer (initiator). Naming schemes are broadly classified as flat names and hierarchical or human friendly names. Using flat names hide the original meaning and context of the object since they are hashed. The drawback of this mechanism is that they are difficult to remember. On the other hand, the hierarchical naming schemes are human friendly because of the fact that their structure is more like that of a URL, describing the content and context of the requested object. Human friendliness makes it easier for users to find the objects they are interested in,

but this brings a few challenges like authenticity, security binding, and ensuring the uniqueness of objects globally [18]. A few recent naming architectures designed to overcome security issues caused by naming schemes are described below:

a. NetInf [19]

The study presents an architecture consisting of a naming scheme that enables verification of data integrity, owner authentication, and identification along with naming the content. The scheme provides a combination of name persistence, self-certification, owner authentication, and owner identification. Name persistence is ensured even if the content specifications like owner or location change. Self-certification makes it easier for the user to verify the data integrity without requiring trust in any third party or the producer of the content. It is usually provided by joining the message/content digest with the original message. Owner authentication and owner identification binds an ID to an entity/content, which enables its verification at the consumer end. This is provided by keeping the encryption key pairs of the owner separate from the self-certification key pairs (which authenticate the content of the data). In NetInf, any entity/content is represented by a globally unique ID. The prototype of the naming scheme is built in java and is tested in windows, Linux, and android platforms. The naming scheme are implemented in various web browsers and clients as a plug-in.

b. Wang et al. [20]

The secure naming scheme proposed in this paper, can locate resources in generic content centric networks. It is an extension of the existing URL naming scheme. Allowing secure content retrieval from various unknown and un-trusted sources is among the primary goals of this algorithm. The features of this scheme include backward compatibility with URL naming schemes and permission for content identification independently. Backward compatibility is the main feature of this scheme. Moreover, it provides secure content retrieval from varying sources, content authentication, and validation with reference to the source and uninterrupted mobility over the network.

c. Name-based Trust and Security Model [21]

A name-based trust and security approach was proposed in [21], which is a security-oriented modification based on the naming system on CCN. The proposal is constructed on top of the identity-based cryptography, which is based on allotting tokens of identification to the users, and only the authorized identities can access data. The identity token derived from any feature of the user or owner's identity, name of the content or its prefix is also used as the public. Since the signed identity creates a direct link between the name (identity) and the content, a malicious user can access it easily and an attack can be launched [22]. An attacker can be anyone from the ISP to the end-users of the system. Another scheme, [22], is an enhanced version of this scheme, which has replaced the identity-based cryptography used in it with hierarchical identity-based cryptography to overcome its drawbacks. Besides the naming related security issues, we briefly discuss the effect of caching on the security and privacy of generic CCNs [23].

d. Risks due to caching

The feature of caching in generic CCN makes them different from traditional TCP/IP networks as CCN stores all the data in the router caches. However, data leaves its traces even after it is removed or forwarded and can be retrieved by an attacker by probing and other such techniques. Hence, caching puts the system security at risk despite being a defining feature of the CCN. Attacks due to caching of data can be reduced by keeping data strictly up to date and restricting the replay of any old data.

e. DoS attack risk)

Denial of Service (DoS) attacks, although difficult to carry out in these networks are still possible. They are carried out through Interests. Since Data packets of request and response always follow the same path, they can be exploited and a destination can be flooded with irrelevant Data packets. The complications arise because the origin of the packet with respect to its location cannot be identified. Due to the architecture of Internet and the generic CCNs, DoS attackers can easily disguise themselves. In order to cope with these attacks, a few security patches (to be added to the existing system) and trust mechanisms like firewalls and spam filtering have been developed. New security protocols that complement the existing networking protocols are also introduced to solve this problem [24].

f. Security model

Establishing security of data in such a system, which is largely content-oriented, is mandatory before it can be implemented on a large scale. Security has been a relatively less explored area as compared to the caching or mobility of generic CCNs and requires further investigation to determine a better security model.

4. CACHING

A key feature of CCN is in-network caching of content [20]. The CCN idea is based upon showing of *Interest* in content and in turn serving of actual content against the *Interest*. This idea is achieved by allocating a globally unique identifier (GUID) for each content in the network. This GUID can be comprehended by all network nodes including routers. Hence, routers can cache *Data* packets and, therefore, serve future content *Interest* requests via its own cache instead of forwarding *Interests* to server. Due to the increase in content number and variety, caching has become the leading variable for efficient utilization of CCN, and hence an interesting research problem. A number of schemes have already been proposed for targeting caching in CCN [20, 25-27]. The problems of caching in CCN include: which content is to be cached, when is appropriate timing for caching, how would content be cached (storing and eviction) and on which network path a content be cached. Furthermore, there is problem of cache allocation to network routers that states how much space must be allocated to each router. There are two approaches: (1) Homogeneous and (2) Heterogeneous [20]. The homogeneous approach allocates equal cache size to each router while heterogeneous allocates higher space to some routers and lower to others based on some optimal criteria. Some recent research projects on the aforementioned caching problems are analyzed and compared in following sub sections.

a. Cache Allocation Approaches

The authors in [20] presented an algorithm to compute optimal cache allocation among routers. The algorithm focuses on maximization of aggregate benefit where benefit means reduction in hops taken by an *Interest* packet from client to server. The *Interest* packet is assumed to follow shortest path from client to server. Therefore, a shortest path tree rooted at the server that has the desired data for clients is formed. The shortest path tree structure leads the problem of cache allocation to be divided in two sub problems: (1) maximizing benefit of a *Data* packet that has highest probability of being requested in a shortest path tree; and (2) maximizing overall benefit of all *Data* packets in whole network that is sum of all maximizations in previous step. The former problem is being solved with help of k-means algorithm while a greedy approach is adopted for the latter. The proposed greedy algorithm produces a binary matrix having nodes in rows and content packets in columns, depicting content allocations on nodes. The complexity of proposed optimization algorithm is *max(O(sn3), O(ctotallogN))*, where *s* is number of servers, *n* is total number of nodes, *N* is number of contents, and ctotal is total cache space of network. Experimental evaluation showed that cache allocation should be heterogeneous and related to topology and content popularity.

b. Data Allocation Approaches

Cache management and content request routing policies have been proposed in [28]. Cache management policy decides whether a piece of data that passes through a router must be cached. If so then the policy determines which portion of data residing in the cache must be removed to make room for data to be cached. In caching and removal process of data, link congestion along the path of data retrieval and data popularity are considered. A utility function is used in the decision of caching a piece of content on a node. The utility function takes minimum bandwidth of content retrieval path and content popularity as parameters. If a piece of content has higher utility value than lowest utility value of any content at a node, the content is cached in place of lowest utility value content. The main idea of cache management policy is that content forwarded over congested links is retained by caches while evicting content forwarded over uncongested links. The content request routing policy utilizes a scoped-flooding protocol. The protocol locates requested cached content with minimum download delay as there are multiple caches having a piece of content. The scope is boundary for the protocol to stop searching for cached content, and is calculated in terms of number of hops that a packet traverses. An issue with such caching policy is that there might be no interested node for a content after some time that is residing with a high utility value in a cache. Such a content is never removed from a cache due to higher utility value.

In [29], the authors proposed a latency-aware cache management mechanism for content centric network. The mechanism is based upon the principal that whenever a content is retrieved, it is stored into cache with a probability proportional to its recently observed retrieval latency. There is a tradeoff between cache size and delivery time as reducing delivery time of every content requires a cache of size appropriate to store every content at every node [29]. The proposed mechanism minimizes this tradeoff by assigning caching priority to contents that are farther to retrieve. It is a fully distributed mechanism since the network caches do not exchange caching information. Two main benefits of the proposed mechanisms are: (1) faster delivery time as compared to latency insensitive approaches (2) fast convergence to optimal caching situation. However, dynamic network conditions have not been considered while calculating optimal data caching.

In [30], a cross-layer cooperative caching strategy for CCN is proposed. Three parameters that are user preference, betweenness centrality and cache replacement rate are introduced as application layer, network layer and physical layer metrics. A caching probability function is derived based on Grey Relational Analysis among all nodes along the content delivery path. Multiple requests for the same content are aggregated in PIT. PIT keeps tracks of *Interest* packets forwarded upstream so that returned *Data* packet can follow the path back to endpoint of content requester. Some nodes aggregate requests and termed as aggregated node. The aggregated node computes the caching probabilities and adds them in the PIT. Caching probability fields of *Data* packet is updated by the caching probability in PIT when the *Data* packet arrives at the aggregated node.

A novel caching method to deliver content over the content centric networks has been proposed in [31]. The authors analyze caching content distribution and related *Interest* distribution by considering the mechanism of CS and PIT of CCNs. Based on the analysis, a caching algorithm has been proposed to efficiently use the capacity of CS and PIT. The caching is done based on a cost model derived in the paper. The cost model is based on total delay to retrieve a piece of content. The CS calculates the cost to remove a cached content for caching a newly requested piece of content. If the cost is below a threshold, the new content is cached. The proposed scheme does not account for the dynamic network conditions and it is a distributed caching mechanism.

The research work in [32] is focused on video streaming problem in Vehicular Ad-hoc Networks (VANETs). Due to dynamic change in bandwidth in VANETs, dynamic adaptive streaming (DAS) technology is used for delivery of video content with different bit rates according to available bandwidth. DAS requires several versions of a same video content, resulting in reduced cache utilization. Hence, a cache management scheme for adaptive scalable video streaming in Vehicular CCNs has been proposed in [32]. The scheme aims to provide high QoE video streaming services through caching of appropriate bit rate content near consumers. A video content is divided into layers where each layer is the video content with specific bit rate. Chunk of layers are pushed to neighbors according to their available bandwidth through broadcast hop-by-hop.

In [33], the authors studied the user-behavior driven CCN caching and jointly investigated video popularity and video drop ratio. It has been said that this is the first paper taking the video drop ratio into consideration in CCN caching policy design. The video drop ration means that video consumer does not always watch the video until the end. An intra-domain caching in CCN system has been considered, that is composed of servers/repositories, routers and users. CS of each router acts as a buffer memory. The routers are divided in levels hierarchy and videos are ranked according to popularity. The main idea of proposed scheme is that *ith* chunk is cached in *jth* router if and only if the higher-ranking chunks than *ith* chunk can be stored in lower level routers than *jth* router. Also if the lower level routers cannot cache all the chunks having higher ranking then *ith* chunk. The proposed algorithm achieves optimal values in terms of average transmission hops needed and server hit rate given a cascade network topology.

A caching policy similar to the one mentioned in [33] has been proposed in [34], which maximizes cache utilization and improves content diversity in networks by content popularity and node level matched based caching probability. Three parameters including hop account to the requester, betweenness centrality and cache space replacement rate are considered for evaluation of caching property of nodes along the delivery path. The proposed strategy performs classification of nodes into levels based on their caching property. High caching property nodes are defined as first level nodes; while less capable nodes are termed as second level nodes. Moreover, the strategy uses a probability-based mechanism to compare and match the content popularity with the matching level of the node. Nodes only cache the content, which match the probability of storing. The scheme warrants caching more popular content at first level of nodes, which decreases the redundancy of less popular content. In the same way, less popular content is cached to next level of nodes. The proposed policy improved up to 23% cache hit ratio, reduced up to 13% content access delay, and accommodated up to 14% more contents compared with Leave Copy Everywhere (LCE) policy.

Table 1 provides a comparative analysis of the CCN caching schemes mentioned in this section based on various critical parameters. Table 2 classify a number of CCN schemes in terms of three important aspects of data delivery: routing, forwarding and caching. We note from Table 2 that none of the schemes provides an integrated solution targeted these three aspects together in one solution.

Table 1. Caching schemes comparison

Ref.	Topologies Used	Caching Objective	Cache Allocation Type	Cache Allocation Criteria	Data Allocation Criteria	Performance Metric	Performance Metric Used
[20]	Barbasi-Albert, Watts-Strogatz	Cache Allocation	Heterogeneous (BA) and Homogeneous (WS)	Topology and Interest dependent	-	Network-centric	Remaining Traffic
[35]	Arbitrary topologies	Cache Allocation	Heterogeneous	Nodes centrality dependent	-	Network-centric and User-centric	Cache Hit Probability and Path Stretch
[28]	Grid, Scale-free, Rocketfuel, Hybrid	Data Allocation	-	-	Links congestion and Interest dependent	User-centric	Content Download Delay
[36]	Binary Tree	Data Allocation	-	-	Hop count from client to cache	Network-centric	Server Hits and Hop Reduction
[37]	Netrail, Abilene, Claranet, Airtel, Geant	Data and Cache Allocation	Heterogeneous	Data caching probability and Budget	Traffic Cost and Flow Minimization	Network and User-centric	Budget and traffic flow
[29]	Line, Binary tree	Data Allocation	-	-	Content Popularity and Latency	Network-centric	Data retrieval latency
[38]	k-ary tree, scale-free	Data Allocation	-	-	Betweenness Centrality of Node	Network-centric	Hop reduction ratio, server hits reduction ratio
[39]	CERNET2	Data Allocation	-	-	Content popularity	User-centric	Network delay
[40]	10x10 grid, 6-level binary tree	Data Allocation	-	-	Ubiquitous	Network-centric	Content travelling average distance
[41]	k-ary tree, scale-free	Data Allocation	-	-	Betweenness Centrality of Node	Network and User-centric	Latency and congestion
[42]	AS 1755, AS 3967, Brite1, Brite2	Data allocation	-	-	Benefit of eviction of a content	Network-centric	Content redundancy and cache hit rate
[43]	5-level binary tree,	Data allocation	-	-	Content popularity	Network and User-centric	Content redundancy and content download delay
[30]	Scale-free	Data allocation	-	-	User preference, betweenness centrality, cache replacement rate	Network-centric	Cache hit ratio, Average hops
[31]	Unspecified	Data allocation	-	-	Network delay cost	User-centric	Network delay
[33]	Cascade topology	Data allocation	-	-	Video popularity and drop ratio	Network and User-centric	Average transmission hops and server hit rate
[34]	Scale-free	Data allocation	-	-	Probability based on nodes level	Network and User-centric	Cache hit ration, Content number, and Content access delay
[44]	Unspecified	Data allocation	-	-	Cache free space	Network and User-centric	Cache hit, Cache miss, and Total number of Replacements

Table 1. Caching schemes comparison (*continue*)

Ref.	Topologies Used	Caching Objective	Cache Allocation Type	Cache Allocation Criteria	Data Allocation Criteria	Performance Metric	Performance Metric Used
[27]	Arbitrary	Cache allocation	Heterogeneous	Power consumption	-	Network and User-centric	Average round trip hops, Power consumption, Number of caching nodes
[45]	Abilene, CERNET, GEANT, US-A	Cache allocation	Heterogeneous	Network performance and cache provisioning cost	-	Network and User-centric	Hop count, Latency
[32]	Manhattan mobility model	Data allocation	-	-	Link bandwidth	Network and User-centric	Average video freeze time and ratio, Average bit rate, Average cache hit ratio

Table 1. Objectives based comparison

Reference	Caching	Routing	Forwarding
[20]	✓	×	-
[35]	✓	×	×
[28]	✓	✓	×
[36]	✓	×	×
[37]	✓	×	×
[45]	✓	✓	×
[29]	✓	×	×
[38]	✓	×	×
[39]	✓	✓	×
[40]	✓	×	✓
[41]	✓	×	×
[42]	✓	×	×
[43]	✓	✓	×
[30]	✓	×	×
[27]	✓	×	×
[33]	✓	×	×
[34]	✓	×	×
[44]	✓	×	×
[30]	✓	×	×
[32]	✓	×	×

c. Quantitative comparisons

We present a quantitative comparison of some of the selected most popular and recent caching schemes to gain an insight into how the variation in network parameters effect the caching performance. The selected caching schemes include: LCE [46], LCD [40], ProbCache [36], CL4M [38], and opt-Cache [39]. These schemes are also most cited in the literature. Here is the brief description of the chosen schemes:

- Leave Copy Everywhere (LCE) [46] stores a copy of each requested content from the server to each node on the path. When a content is to be replaced to accommodate newer, the LCE scheme replace the existing contents in least recently used manner, such that the content that is least used in the recent time is replaced with the newer content.
- Leave Copy Down (LCD) aims to place a copy of content to most closer node possible to the requester. Usually, the content is stored at the immediate neighbor of the requesting node.
- ProbCache performs the in-network caching of content by devising a probabilistic model for storage related decisions of content. The probabilistic assignment of content helps in addressing the cache redundancy issue.
- Cache Less for More (CL4M) aims to conserve the caching space by only storing content on a single node along the delivery path. The concept of betweenness similarity is utilized in the proposed scheme, so that a node residing on the shortest delivery path among all node pairs is considered to be more central for content storage.

- Opt-Cache presents a heuristic approach to optimize the caching performance. The proposed scheme computes an ideal set of contents to be placed on edge nodes by considering objectives such as content popularity and distance from content source.

d. Simulation setup

For simulations, we considered Icarus [47] simulator that is specifically designed for CCN and supports implementation of newer caching policies along with provision of various network topologies and scenarios. The default shortest path algorithm to route Interest packet from client to content source in Icarus is the famous Dijkstra algorithm. Table 3 shows the basic simulation settings. The in-depth analysis of these results is published in [48].

Table 3. Simulation settings

Parameter	Value
Content popularity threshold (α)	0.6, 0.8, 1.0
Number of contents	10K–50K
Cache size (% of buffer)	4% – 20%
Request rate	1.0 request/sec
Topology	GEANT

e. Performance metrics

- *Latency* (in milliseconds): It refers to the delay incurred in delivering the requested content to the end user. The latency is computed as:

$$Latency = request\ travel\ delay\ +\ content\ travel\ delay \quad (1)$$

- *Hit Ratio:* This measures the number of times the requested content was actually present on the node where the request arrived. The hit ratio is computed using following formula:

$$Hit\ Ratio = \frac{\#Cache\ hits}{\#Cache\ hits + \#Cache\ miss} \quad (2)$$

f. Latency performance with varying cache size

Figure 3(a)-(c) shows the latency performance graphs of the selected schemes for the content popularity parameter values of α=0.6, 0.8, and 1.0, respectively. For these graphs, the number of contents is kept constant to be 1,000. It can be seen that LCE exhibits maximum latency despite being a data redundant scheme. This is because the LCE do not consider the content popularity and future access probability while making the storage decision. The latency performance of ProbCache is better than CL4M, as the former has a better prediction model to compute the future content access probability, thus reducing the redundancy, whereas the latter address redundancy by caching content on more central node, which may not be ideally closer to the end users. Since, LCD scheme caches content on the delivery path to the end user, it shows lesser latency than other schemes, but this comes at a tradeoff of data redundancy. Finally, due to consideration of both the path lenght and the content popularity features, Opt-Cache shows comparatively minimum latency, as frequently accessed popular content is placed closer to the end users.

g. Latency performance with varying number of contents

Figure 4(a)–(c) indicate the performance of selected schemes by varying number of contents for the content popularity parameter values of α=0.6, 0.8, and 1.0, respectively, with cache size kept constant at 20%. It can be observed that the performance of the schemes is almost consistent to the latency graphs in Figure 3, due to the same reasons illustrated before. The latency has a downward trend with increase in the number of contents as the contents are more readily available, being cached in nearby nodes, for new requests. The latency also appears to be sensitive to content popularity parameter, as the content popularity parameter value increases, the latency further decreases, as the popular content is readily accessible for incoming requests.

h. Hit Ratio performance with varying cache size

Figure 5(a)–(c) display the hit ratio performance of the selected schemes by varying the cache size for the content popularity parameter values of α=0.6, 0.8, and 1.0, respectively. It can be seen from the graphs that the hit ratio performance of almost all the schemes improve with the increase in the cache size. The ProbCache and Opt-cache show better performance for different values of α compared to the remaining schemes due to ideal placement of content over the network.

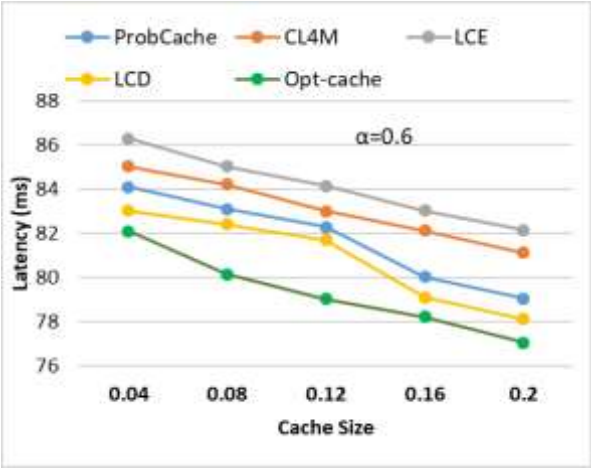

Figure 3(a). Latency performance with varying cache size and α=0.6

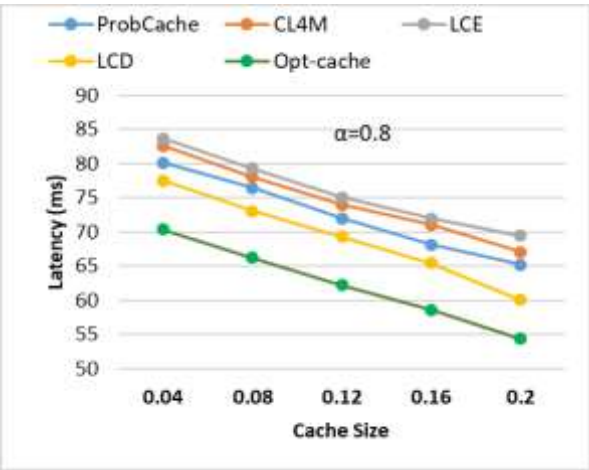

Figure 3(b). Latency performance with varying cache size and α=0.8

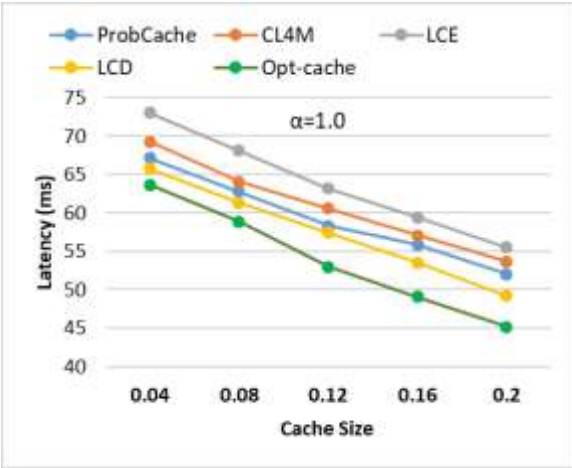

Figure 3(c). Latency performance with varying cache size and α=0.8

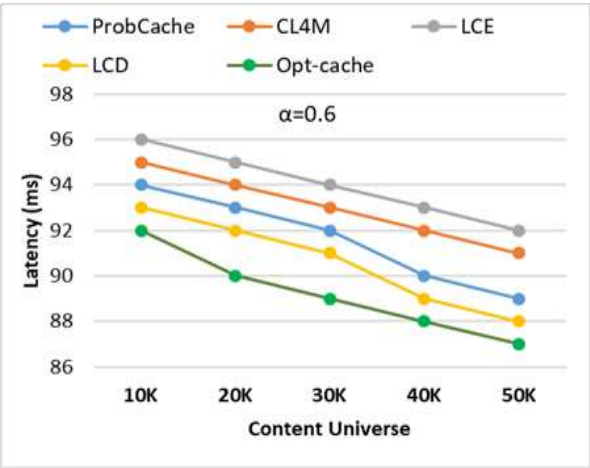

Figure 4(a). Latency performance with varying content count and α=0.6

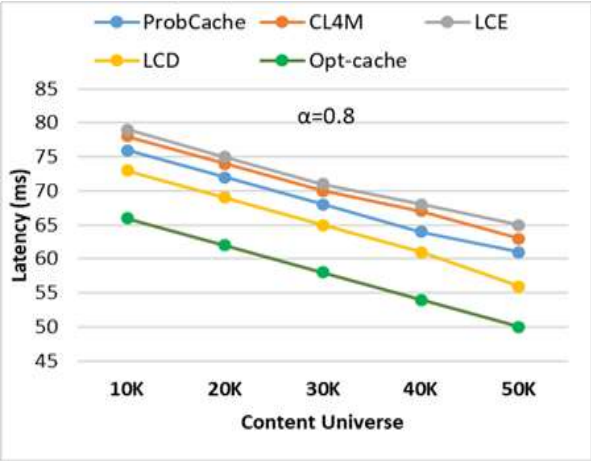

Figure 4(b). Latency performance with varying content count and α=0.8

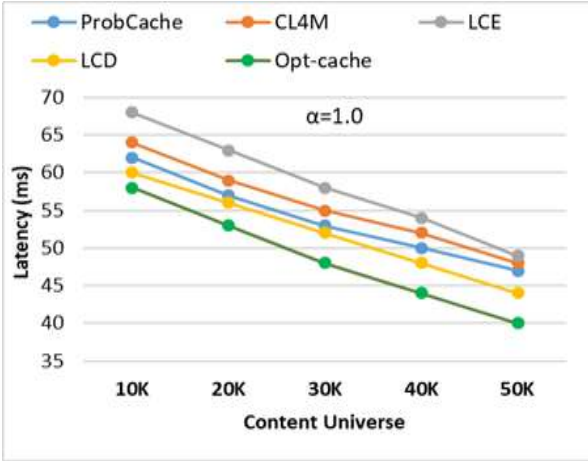

Figure 4(c). Latency performance with varying content count and α=1.0

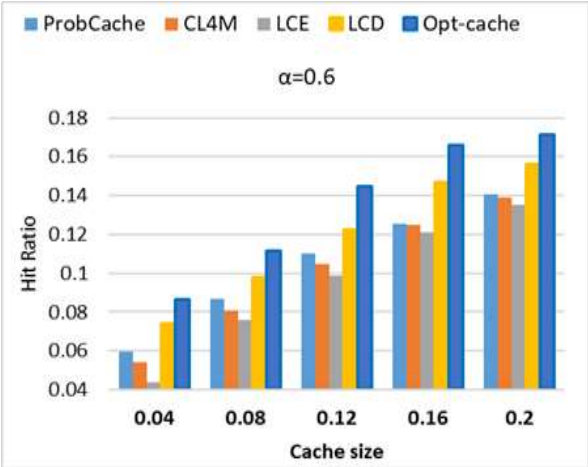

Figure 5(a). Hit ratio with varying cache size and α=0.6

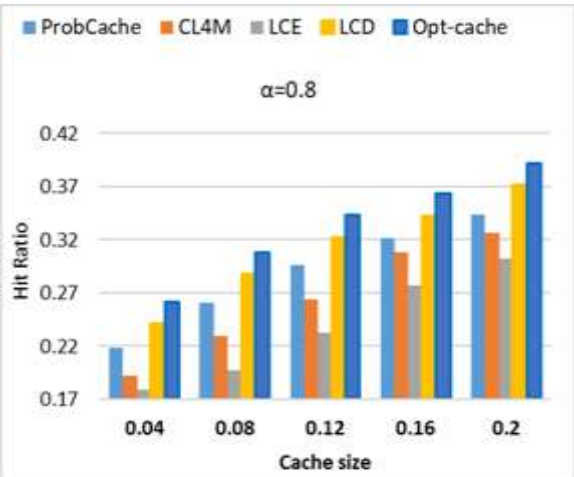

Figure 5(b). Hit ratio with varying cache size and α=0.8

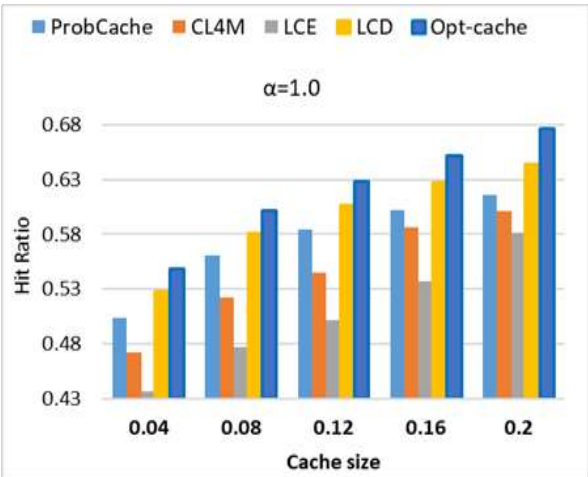

Figure 5(c). Latency performance with varying content count and α=1.0

5. CONCLUSION

This paper explores the features of CCN, looking at how the architecture could support mobility, caching and security elements. Different management techniques of each of these challenges have been discussed and analyzed in detail. The caching of contents at local level replication is a challenge in CCN architecture. Hence, caching schemes discussed in this paper are classified in two types: (1) Data allocation, (2) Cache allocation. Data allocation issue has been addressed by simpler techniques like centrality based or by complex probabilistic techniques. Moreover, routing and forwarding techniques are being coupled with caching schemes by some of the approaches to enhance network performance. The cache allocation problem is classified further into homogeneous and heterogeneous cache allocation to the routers.

Most of the researchers prefer heterogeneous while some argue that heterogeneous policies offer less gain than homogeneous allocations. Based on caching schemes analysis, we conclude that there are many parameters such as network topology, mobility, nodes capacity, and content popularity that effect caching scheme performance. Every scheme achieves some gains based on the prior assumptions on those parameters. Hybrid caching schemes could be designed but in terms of complexity tradeoff on network routers. Moreover, it is observed that most of the CCN schemes do not provide integrated solution, which targets caching, routing and forwarding together. We also found that security has to be the focus of research in CCN and needs to be further explored. Hence, CCN security is a significant challenge and an open research area.

REFERENCES

[1] MS Akbar, KA Khaliq, RNB Rais, A Qayyum, “Information-centric networks: Categorizations, challenges, and classifications,” in *Proc. of 2014 23rd Wireless and Optical Communication Conference (WOCC)*, pages 1-5, New Jersey, USA, 2014.

[2] Cisco Visual Networking Index: 2017-2022 White Paper, https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html. Accessed: Oct 2019.

[3] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, R. L. Braynard, “ Networking Named Content”, ACM CoNEXT 2009, Rome, December, 2009.

[4] G. Xylomenos *et al*., "A Survey of Information-Centric Networking Research," in *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024-1049, Second Quarter 2014.

[5] M. Gohar, N. Khan, A. Ahmad, M. Najam-Ul-Islam, S. Sarwar, and S-J Koh, “Cluster-Based Device Mobility Management in Named Data Networking for Vehicular Networks,” *Mobile Information Systems*, vol. 2018, 2018.

[6] C. Tsilopoulos and G. Xylomenos, “Supporting Diverse Traffic Types in Information Centric Networks”, in *Proc. of Information Centric Networking Conference*, Toronto, Canada, Aug 2011.

[7] D. Kim, J. Kim, Y. Kim, H. Yoon, and I. Yeom. "Mobility support in content centric networks," in *Proc. of the ACM 2nd ICN workshop on Information-centric networking*, 2012.

[8] J. Lee, D. Kim, M. Jang and B. Lee, "Proxy-based mobility management scheme in mobile content centric networking (CCN) environments," *2011 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, 2011, pp. 595-596.

[9] S. Naz , R.N.B. Rais , P.A. Shah, S. Yasmin, A. Qayyum, S. Rho and Y. Nam, “A dynamic caching strategy for CCN-based MANETs,” *Computer Networks*, vol. 142, pp. 93–107, 2018.

[10] J. Lee, S. Cho and D. Kim, "Device mobility management in content-centric networking," in *IEEE Communications Magazine*, vol. 50, no. 12, pp. 28-34, December 2012.

[11] J. Kuriharay, E. Uzun and C. A. Wood, "An encryption-based access control framework for content-centric networking," *2015 IFIP Networking Conference (IFIP Networking)*, Toulouse, pp. 1-9, 2015.

[12] C. A. Wood and E. Uzun, "Flexible end-to-end content security in CCN," *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, pp. 858-865, 2014.

[13] W. Walter and P. Nikander "Secure naming in information-centric networks," *in Proc. of the ACM Re-Architecting the Internet Workshop*, 2010.

[14] C. Ellison and B. Schneier, “Ten Risks of PKI: What You’re Not Being Told About Public Key Infrastructure,” *Computer Security Journal*, 16(1), 2000.

[15] D. Smetters and V. Jacobson, “Securing network content,” *PARC*, Technical Report, Oct 2009.

[16] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker, “Naming in content-oriented architectures,” *in Proc. of ACM Workshop on Information-Centric Networking (ICN)*, 2011.

[17] M. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba and B. Mathieu, "A survey of naming and routing in information-centric networks," in *IEEE Communications Magazine*, vol. 50, no. 12, pp. 44-53, December 2012.

[18] A. Ghodsi, “Naming in Content-Oriented Architectures,” *in Proc. of ACM SIGCOMM Workshop on Information Centric Networking*, 2011.

[19] C. Dannewitz, J. Golic, B. Ohlman, and B. Ahlgren, “Secure Naming for a Network of Information,” *in Proc. of IEEE Conference on Computer Communications Workshops*, 2010.

[20] Yonggong Wang, Zhenyu Li, G. Tyson, S. Uhlig and G. Xie, "Optimal cache allocation for Content-Centric Networking," *2013 21st IEEE International Conference on Network Protocols (ICNP)*, Goettingen, pp. 1-10, 2013.

[21] X. Zhang, K. Chang, H. Xiong, Y. Wen, G. Shi and G. Wang, "Towards name-based trust and security for content-centric network," *2011 19th IEEE International Conference on Network Protocols*, Vancouver, BC, pp. 1-6, 2011.
[22] B. Hamdane, A. Serhrouchni, A. Fadlallah and S. G. E. Fatmi, "Named-Data security scheme for Named Data Networking," *2012 Third International Conference on The Network of the Future (NOF)*, Gammarth, pp. 1-6, 2012.
[23] A. Chaabane, E. D. Cristofaro, M. A. Kaafar, and E. Uzun, "Privacy in Content-Oriented Networking: Threats and Countermeasures," *ACM SIGCOMM Computer Communication Review*, July 2013.
[24] P. Stuckmann and R. Zimmermann, "European research on future Internet design," in *IEEE Wireless Communications*, vol. 16, no. 5, pp. 14-22, October 2009.
[25] S. Naz, R.N.B. Rais, A. Qayyum, "A Resource Efficient Multi-dimensional Cache Management Strategy in Content Centric Networks", *Journal of Computational and Theoretical Nanoscience*, Vol. 15, Issue 4, pp. 1137-1152, 2018.
[26] S. Naz, R. N. B. Rais and A. Qayyum, "Multi-Attribute Caching: Towards efficient cache management in Content-Centric Networks," *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, pp. 630-633, 2016.
[27] L. Zhou, T. Zhang, X. Xu, Z. Zeng and Y. Liu, "Generalized dominating set based cooperative caching for content centric ad hoc networks," *2015 IEEE/CIC International Conference on Communications in China (ICCC)*, Shenzhen, pp. 1-6, 2015.
[28] M. Badov, A. Seetharam, J. Kurose, V. Firoiu, and S. Nanda, "Congestion-Aware Caching and Search in Information-Centric Networks," *in Proc. of ACM ICN*, 2014.
[29] G. Carofiglio, L. Mekinda and L. Muscariello, "LAC: Introducing latency-aware caching in Information-Centric Networks," *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, Clearwater Beach, FL, pp. 422-425, 2015.
[30] L. Wu, T. Zhang, X. Xu, Z. Zeng and Y. Liu, "Grey relational analysis based cross-layer caching for content centric networking," *2015 IEEE/CIC International Conference on Communications in China (ICCC)*, Shenzhen, pp. 1-6, 2015.
[31] S. Zhou, F. Dongfeng, and H. Bo, "Caching Algorithm with a Novel Cost Model to Deliver Content and Its Interest over Content Centric Networks," *China Communications*, pp 23–30, 2015.
[32] Y. Wei, C. Xu, M. Wang and J. Guan, "A novel dynamic adaptive video streaming solution in content-centric mobile network," *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Valencia, pp. 1-7, 2016.
[33] Z. Liu, Y. Ji, X. Jiang, and Y. Tanaka, "User-behavior Driven Video Caching in Content Centric Network," *in Proc. of ACM ICN'16*, pp 197–198, 2016.
[34] Y. Li, T. Zhang, X. Xu, Z. Zeng and Y. Liu, "Content popularity and node level matched based probability caching for content centric networks," *2016 IEEE/CIC International Conference on Communications in China (ICCC)*, Chengdu, pp. 1-6, 2016.
[35] D. Rossi and G. Rossini, "On sizing CCN content stores by exploiting topological information," *2012 Proceedings IEEE INFOCOM Workshops*, Orlando, FL, pp. 280-285, 2012.
[36] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic In-Network Caching for Information-Centric Networks," *in Proc. of ACM ICN*, pp 50-60, 2012.
[37] M. Mangili, F. Martignon, A. Capone, and F. Malucelli, "Content-Aware Planning Models for Information-Centric Networking," *in Proc. of Globecom-Next Generation Networking Symposium*, pp. 1854-1860, 2014.
[38] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache Less for More" in Information-Centric Networks," in Part I, LNCS 7289, R. Bestak, eds., *IFIP International Federation for Information Processing*, pp. 27–40, 2012.
[39] Z. Ming, M. Xu, and Dan Wang, "Age-based Cooperative Caching in Information-Centric Networking," *in Proc. of 23rd International Conference on Computer Communication and Networks (ICCCN)*, 2014.
[40] G. Rossini and D. Rossi, "Coupling Caching and Forwarding: Benefits, Analysis, and Implementation," *in Proc. of ACM ICN*, 2014.
[41] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Coupling Caching and Forwarding: Benefits, Analysis, and Implementation," *in Proc. of ACM ICN*, pp 127–136, 2014.
[42] J. M. Wang, J. Zhang, and B. Bensaou, "Self Assembly Caching with Dynamic Request Routing for Information-Centric Networking," *in Proc. of ACM ICN*, pp 61-66, 2013.
[43] Y. Li, Y. Xu, T. Lin, G. Zhang, Y. Liu, and S. Ci, "Self Assembly Caching with Dynamic Request Routing for Information-Centric Networking," *in Proc. of Globecom - Next Generation Networking Symposium*, pp 2158–2163, 2013.
[44] G. P. Mishra and M. Dave, "Cost Effective Caching in Content Centric Networking," *in Proc. of 1st International Conference on Next Generation Computing Technologies (NGCT-2015)*, pp 198–202, 2015.
[45] Y. Li, H. Xie, Y. Wen, C. Chow and Z. Zhang, "How Much to Coordinate? Optimizing In-Network Caching in Content-Centric Networks," in *IEEE Transactions on Network and Service Management*, vol. 12, no. 3, pp. 420-434, Sept. 2015.
[46] Y. Li, T. Lin, H. Tang, P. Sun, "A Chunk Caching Location and Searching Scheme in Content Centric Networking," in Proc. of IEEE ICC - Next-Generation Networking Symposium, pp. 2655-2659, 2012.
[47] L. Saino, I. Psaras, and g. Pavlou, "Icarus: a caching simulator for information centric networking (ICN)," *in Proc. of the 7th International ICST conference on Simulation Tools and Techniques*, pp. 66-75, 2014.
[48] F. Qazi, O. Khalid, R. N. B. Rais, I. A. Khan, and A. R. Khan, "Optimal Content Caching in Content-Centric Networks," Volume 2019, Article ID 6373960, 15 pages, https://doi.org/10.1155/2019/6373960, 2019.