Integrated Caching and Routing Strategy for Information-Centric Networks

Submitted by:

Naheed Nazneen Tuli

2016-2-98-002

Department of Electronics and Communication Engineering
East West University

Summer 2017

Supervised by:

Dr. Mohammad Arifuzzaman

Assistant Professor

Department of Electronics and Communication Engineering

East West University

Declaration

I hereby declare that this research project report is an original piece of work carried out by me under the guidance and supervision of **Dr. Mohammad Arifuzzaman**. This report is the requirement for the successive completion of M.Sc.in Telecommunication Engineering under the department of Electronics and Communication Engineering.

I state that the report along with its literature that has been demonstrated in this report papers, is our own work with the masterly guidance and fruitful assistance of our supervisor for the finalization of our report successfully.

Signature:

Naheed Nazneen Tuli

ID: 2016-2-98-002

Signature of Supervisor:

Dr. Mohammad Arifuzzaman
Assistant Professor
Department of Electronics and Communications Engineering,
East West university.
Dhaka, Bangladesh.

Approval

This Research Project report on **"Integrated Caching and Routing Strategy for Information-Centric Networks"** has been submitted to the following board of examiners as a partial fulfillment of the requirements for the award degree Masters of Science in Telecommunication Engineering under the department of Electronics and Communication Engineering on August 2017 by the following student has been accepted as satisfactory.

Naheed Nazneen Tuli

ID: 2016-2-98-002

Signature of supervisor:

Dr. Mohammad Arifuzzaman
Assistant Professor
Department of Electronics and Communications Engineering,
East West university.
Dhaka, Bangladesh.

Approved By:

(Dr.M. Mofazzal Hossain)
Professor & chairperson
Department of Electronics and Communication Engineering
East West University, Dhaka, Bangladesh

Acknowledgement

I would like to express our gratitude and appreciation to all those who gave me the possibility to complete this research work. A special thanks to my Supervisor **Dr. Mohammad Arifuzzaman** whose help, suggestions and encouragements helped me to take my thesis especially on Caching. He supported me by showing different method of information collection while doing this work. He always helped me when required and he gave us right direction towards completion of this work.

I also want to thanks all faculty and staff of Department of Electronics and Communication Engineering of East West University for their full cooperation during the period of the report completion, from the beginning till end. Also thanks to all of my family members, friends and everyone who contributed by supporting our research work and helped me during the research progress till it is fully completed.

Abstract

Besides Off-path caching in Information Centric Networking (ICN), On-path caching is an integrated caching solution with-in an Autonomous System's (AS) local network. However, it is seen that in On-path caching, the content is cached en-route in the reverse path towards the Interest generator. Thus, local Rendezvous Network (RENE)[29] /Name Resolution System (NRS) [15] (in routing through name resolution) or FIB [2] (in name-based routing) is not conscious of the cached data. As a result the most extensively deployed intra-domain routing protocol and their forwarding strategy is not capable of addressing all available temporary cached copies of content. This leads on-path caching strategy to experience from two major downsides: numerous replicas of same content within AS which do not add significant value in terms of cache resource utilization and a trial taken to minimize redundancy ends up with the cost of fetching more copy of similar content from the repository (i.e., the closest cache copy is unknown/not enroute).

In this paper we introduce an integrated caching and forwarding solution which minimizes the caching redundancy and maximizes the probability of finding nearby cached content, which causes utilization of available cache resources of AS's as a whole to be productive. We propose novel concepts of naming the content router and caching the content's cache-route to address the issue. We leverage Name Data Networking (NDN) node architecture and its forwarding plane/strategy (of routing) to attain our goal. Mathematical analysis to determine load balance and cache hit ratio, and simulation results considering average hop-distance and server hit ratio authenticate the efficiency of our proposed scheme.

Table of Contents

List of Figures and Tables

Abbreviation	Elaboration
ICN	Information Centric Network
AS	Autonomous System
RENE	Rendezvous Network
NRS	Name Resolution System
NDN	Named Data Networking
CCN	Content Centric Network
PSI	Publish-Subscribe Internet
DAN	Data Aware Networking
NDO	Named Data Object
LCE	Leave Copy Everywhere
LCD	Leave Copy Down
CS	Content Store
PIT	Pending Interest Table
FIB	Forwarding Interest Base
LRU	Least Recently Used
TLV	Type-length-value
RIB	Routing Information Base
TLS	Transport Layer Security

Table 1: Summary of abbreviations

Chapter 1

Introduction

Chapter 1 Introduction

1.1 Background

Information-Centric Networking (ICN) has captured prominent attention in recent years [1-8]. In PSIRP [1], Content-Centric Networking (CCN) is used as a base to interpret a new architecture. Identifiers are described in an information-centric manner. The Publish-Subscribe Internet (PSI) architecture focuses on content instead of end point communication. PSI wants to change the current Internet architecture that targets on messages and depend completely on senders, to an Internet architecture that replies on information and the receiver. This variation in architecture is required to overcome the disadvantages of complex, inflexible and inefficient network to scalable multicast and highly efficient network.

In [2], authors illustrate CCN that considers content as a primitive – decoupling location from identity, security and access, and retrieving content in terms of name. A prototype CCN network stack is implemented by authors where they demonstrated its necessity for both content distribution and point-to point network protocols. CCN focuses from content location to content names. This is illustrated in Figure 1. In [3], one of the major key problem spaces of Data Aware Networking (DAN) is 'Mobility' and authors propose Named-Node-Networking as a novel architecture for DAN.

In this paper, we deploy CCN node architecture, Named Data Networking (NDN) and forwarding engine strategy to overcome the problem of too many copies of the same content that leads to low cache resource utilization and an increase in overall latency of the network. The main objective is to cache objects only in one router thus leaving space for other objects or popular information to be cached when needed. If the object is cached in all routers, although it has an advantage of achieving the requested object easily, the memory falls short and the cache files are replaced several times whenever a new request comes. On the other hand, if the object is cached in some of the routers, it is difficult to meet the request. With each time a request occurs, if it is not found in the path, the request is forwarded to the repository. This paper addresses routing and caching strategy together so that it minimizes the number of replicas within the AS and at the same time minimizes server hit ratio.

According to the structure of architectural design of ICN, it is evident that the advantages of ICN depend extensively on the widespread cache structure. The main purpose to use widespread cache structure is to achieve reduced latency which is a key feature for current and future internet**.** In 2002, “En-route caching” or “transparent” structure of caching was developed [9]. The term “en-route cache” intercepts a client's request that passes through it. It looks for the requested object in the cache, if the object is in the cache it will be sent back to the client. Thus the request will no longer ponder along the path. Or else, the request will be forwarded along the regular routing path. There are several benefits of en-route caching, which includes: 1) it is transparent to both content server and clients 2) none of the request is detoured off the usual path which reduces the network delay for cache miss and extinguishes extra overhead such as broadcast queries. This en-route caching is one of the important caching strategies of ICN and also a focusing area of this article.

The rest of the paper is organized as follows: In Section 2, we focus on some related works of ICN. We leverage NDN node architecture, naming concept and the forwarding engine model; which are described in Section 3. In Section 4, our proposed caching and forwarding strategy are elaborated. In Section 5, we present the results and discussions of our proposal. Finally, section 6 concludes the paper with a summary and some scope of future work.

1.2 Related Work

Recently, some of the proposals scrutinize the economic incentives in ICN [4], [5]. In [4], authors explain the importance of the socio-economic issues when analyzing the future Internet architecture with respect to public policy. ICN's capability to resolve various complications between different constellations of stakeholder's interests, conflicts etc. are also taken into account. In [5], authors recommend an engineering economic model for ICN by exploiting game theory. At first, the two player game model are represented between Publisher and Telco CDN .Then, improvement of game is done among three major network players (Publisher, Global

CDN and Telco CDN) of ICN. This is done to examine the economic incentives sharing among them. The paper also focuses the requirement for a common standard for content routers (CR) so that as a node in the ICN, CR will ensure scalable content delivery .The procedures of diversion of different network players in ICN required to set up distributed storage architecture are extensively evaluated. To enhance the service delivery in networking [6], smart grid [7], intelligent transport system [8], etc., ICN scheme and architectural model can be utilized.

The research works of on-path caching strategy in ICN can be divided into two broad categories. In the first approach, the content is cached in all routers on the path from content source to destination (request generator) which is proposed in [2]. This is also the usual mode of operation currently in use in most multi-level caches [10]. In many cases, caches are found in multiple levels. Such as: multi-level cache architectures in modern CPUs, multi-level caches in RAID disk arrays, and multi-level caches in the World Wide Web. The requests at first arrives at the lowest level cache (the one closest to the client), and then routed upwards until they reach a cache that stores the desired object. When a request arrives and a hit occurs at a specific level cache or in the origin server, a copy of the requested object is cached in all intermediate caches on the path from the location of the content source (permanent or temporary) down to the request generator. This type of caching technique is named as Leaving Copies Everywhere (LCE). Here, when few Interests come from different points of the network (for a specific content), an identical content is copied on numerous routers. In case of popular contents, this procedure clearly increases the probability of finding content en-route from requester to content repository. Hence, the higher number of copies of the same content diminishes the probability of additional fetching of the same content, as long as it is in the proximity.

On the contrary, the LCE method provides too many duplicates of the same content within AS's local network, without adding substantial value to the AS in terms of cache resource utilization. LCE scheme is criticized due to its content redundancy and inefficiency in resource utilization in the works [11], [12]. Since caches have limited memory, too many duplicates of the popular content results in less amount of space for other contents. A reduction in the number of dissimilar contents cached in the locality (due to the redundant replica of popular contents), minimizes the chance of finding an independent content in the neighborhood as a whole. Also, in LCE scheme, the popularity of the content is not considered during content caching.

The second approach of caching found in the state-of-the-art ICN literature is as follows: here content is copied at one or few routers (chosen randomly or strategically) on the path from content source to destination during content fetching. Works of ProbCache [11], MultiCache [13], Leave Copy Down (LCD) [10] etc., elaborate this category and serve as an example of it. ProbCache is an in-network caching scheme that estimates the ability of caching in a path and it caches contents probabilistically. This provides extra space for other flows in the path and fairly multiplexes contents among caches. MultiCache depends on two primitives, they are multicast and caching. Overlay multicast is used for content delivery which provides various advantages. The purpose of these studies is to reduce redundancy/replica in the neighborhood, which results in ending up with the cost of increasing probability of fetching additional copy of the same content from the repository. In other words, the content is in the locality but it is not en-route. As a result, it is not possible to retrieve the content from the nearby cache since the location of the closest cache copy is unknown [14].

Our projected scheme is to lower the resource utilization (second approach i.e., content caching on one or few routers rather than all routers in the path), as well as support the corresponding forwarding scheme to locate and retrieve closer cache copy rather than searching in server/repository.

There are two different ways to do routing in ICN, they are: Routing through Name Resolution (2-step approach) and, Name-Based Routing (1-step approach). In Name Resolution (ex. SAIL [15]), client directs a name resolution request to the local Name Resolution System (NRS) (if match for the request is not found, it is forwarded to global NRS). NDO (Named Data Object) locator is given back to the client by local NRS. The client then seeks the data source for NDO. The downside of this strategy includes point-of-failure at NRS and a large storage needed by the NRS to store NDO mapping. It is important to take into account that SAIL also supports Name-Based Routing model where NDO information is stored and scattered in network routers using a routing protocol and NDO request can be directly forwarded to the NDO source. Similar to SAIL, PURSUIT also supports both routing approach: Name Resolution by using Rendezvous Function [16] and Name-Based Routing by using topology and forwarding functions. COMET [17] also has provision of both Name Resolution and Name-Based Routing.

The mechanism of Name-Based Routing is to forward the Interest (NDO request) by CCN router to other routers or repositories based on NDO name. NDN uses prefix-based longest match lookups to locate nearby data source (temporary copy or permanent copy of data). The following section will shortly describe the procedure of NDN. CONVERGENCE [18] also uses the Name-Based Routing approach. It uses FIB and RIB rather than using FIB and PIT of NDN. The foundation of our proposed strategy is Name-Based Routing.

In most of the cases, ICN literature works with routing (forwarding) and caching individually and take each of them (routing or caching) as independent problem. It is comprehensibly evident that the accomplishment of on-path caching strategy is determined by the forwarding strategy. In this paper, we introduce an efficient, effective and integrated solution of caching and forwarding in ICN. To the extent of our knowledge, our proposed strategy is the first work, where it is possible to locate a cache copy by forwarding the Interest towards a single shortest path even though the copy is not en-route. Therefore, we take a holistic approach to revisit and address the caching and forwarding strategy of ICN together.

1.3 Information-centric networks

The purpose of ICN [19] is to move from the existing complex Internet model to an easy and generic one. The ICN networking activities depend on the named content objectives and it is a host centric networking model, where end-users only express their interests for a given content. The whole network is in charge of routing the requests based only on the content names towards the best content containers and delivering the contents through the reverse paths to the end-users. The purpose of ICN is to develop the characteristics of networking design. It natively includes the features as location-independent naming, name-based routing, in-networking caching, native multicast, self-secured content, etc.

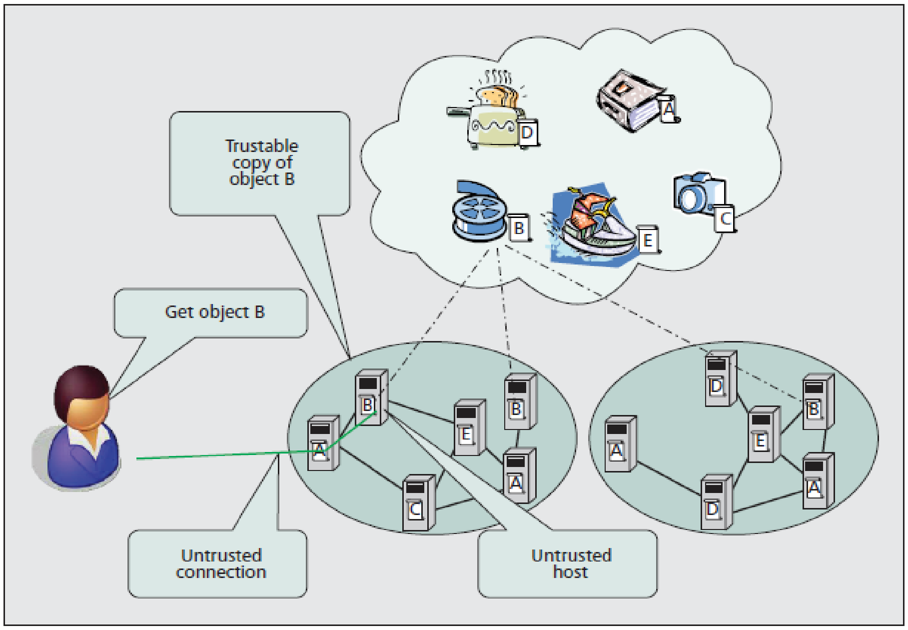

Figure 1. ICN communication model: client side.

Even though ICN enter now into the main stream of networking research, it is still in its early stage. In the last couple of years several projects and research projects have been carried on to recommend a concrete ICN solution to deploy it in real life. The common components in various ICN designs are naming and routing schemes, in-network caching technique and security issues.

Content retrieving in ICN can be mainly divided into two parts: the content discovery and the content delivery. The content discovery deals with how content is named, how it is published and how an ICN node addresses it and the content delivery describes the ICN routing protocol. This includes how a content provider propagate its contents into the network, how an ICN router routes the end-users' interests to the best content sources and how an ICN router deliver the contents to the end-users. In this section we will globally present the ICN naming and routing aspect. The only identifier of each content object the content name that allows the end user or intermediate networking unit to locate the best content holder. The content name usually is a globally unique identifier, but the unique named content can sojourn in different containers, for example the origin contents servers, the CDN repositories or the on-path caches.

1.3.1 Existing solutions of ICN

The TRIAD project [20] is the first one to introduce named content and content name-based routing (Cheriton and Gritter, 2000). After that other researchers demonstrate their interests into this new Internet paradigm and projected several different ICN projects. Brief view of some advanced ICN research project and their naming and routing solutions is described in the following.

1. Content-centric networking (CCN)

Palo Alto Research Center (PARC) proposed the content-centric networking (Jacobson et al., 2009) and the Named Data Networking project (Zhang et al., 2010). It is one of the main attractions of ICN research project. The main objective of CCN is to replace the IP based Internet with a named content based model.

The hierarchical structure of CCN design is shown in (Fig. 1). The name is arranged as a prefix-suffix order. An example of CCN content name is ccnx:/parc.org/video/widget1/version2/chunk2. All the content provided by parc can share the same ccnx:/parc.org/ prefix. The tree-like structure can create the CCN name support the aggregations as the IP address aggregation. A same information object may have several different versions and a single content can be separated into multiple small segments (chunks) in order to adjust with the transport layer. Therefore each CCN name is ended by the version and chunk information which can simplify the content discovery. The entire name is signed with a SHA256 digital signature of the content provider so only the authorized receivers can decrypt the digital data.

The information exchange i n CCN is of two types of packet: the Interest and Data. The Interests can be said as the content which the end-users want to retrieve and the Datas are the response packets which hold the real binary contents. The Interest routing is based on the Content Name. Each CCN node has a component so-called Forwarding Information Base (FIB). The CCN FIB (Fig. 2) is as like as the IP FIB. It holds the Interest routing information. If a content provider wants to publish some contents into the network, it distributes the advertisements into the network. These advertisements will fill the CCN FIB together with the incoming faces as the Interest outgoing faces. Each CCN node can aggregate the FIB entries on the prefix.When the CCN node desires to route an Interest request, it will search the Interest Content Name up in the FIB table. And the Interest will be sent out through the faces of the longest match FIB entry. By default the CCN routing supports the multicast that means one FIB entry may contain more than one outgoing face. This is because one content can be provided by different providers, such as, from the origin content providers, from a CDN repository or from an ISP border-cache.

Another structure of CCN node is Pending-Interest Table (Fig. 2). When a CCN node receives an Interest, if it locally does not have the right content, it should forward the Interest out according to its FIB. Meanwhile, it appends also the Interest name (the ContentName) in its PIT together with the incoming face identifier of the Interest like a "breadcrumb".

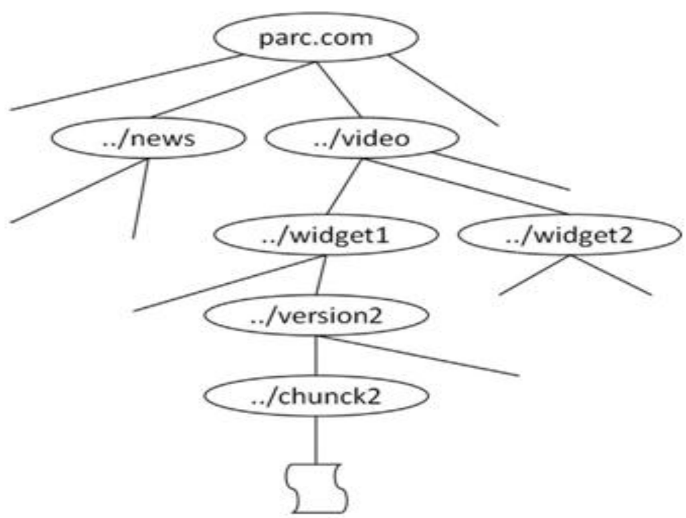

Figure 2. The naming of content-centric network.

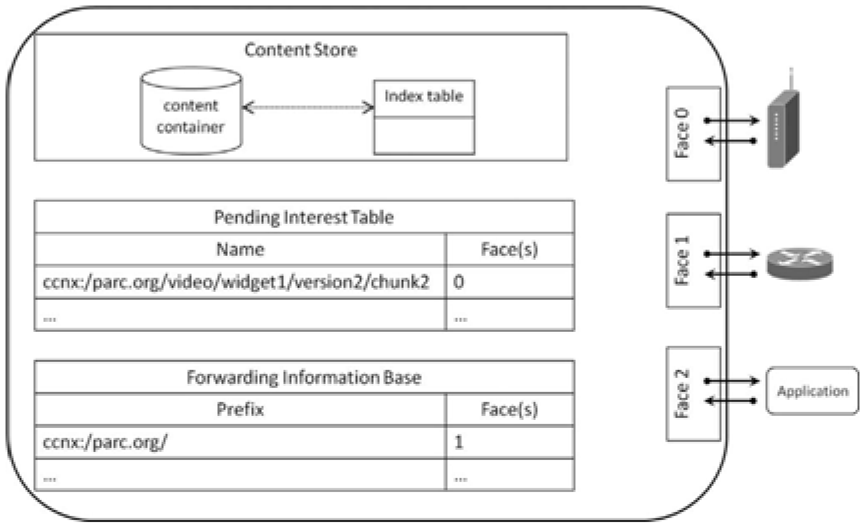

Figure 3. The node structure of content-centric network.

2. Named of information (NetInf)

NetInf (Dannewitz, 2009; Dannewitz et al., 2010a) is a part of the European FP7 project 4WARD (4WARD, 2010). The NetInf recommends an ICN networking architecture which is based on the content register scheme.

The NetInf applies the Information Object (IO) and Bit-level Object (BO) to find the difference between the content identifiers and the real binary content in a Named Data Object (NDO). The naming of NetInf is included in the IOs. The NetInf IO has three parts: the content-identifier, the metadata and the security attributes. The NetInf name uses a flat structure as P:L, thatcontains and splits the identifier of the content provider and the content self. Here, the P is the content provider identifier that is usually the hash of the public key of the provider. The L is the label of the content chosen by the content provider; it is usually the hash of the content itself. The routing of NetInf is a multi-level DHT (MDHT) based registration method which includes a name resolution service (Fig. 4). The MDHT structure contains a three level topology: the Access Node level (AN), the Point of Presence level (POP) and the Autonomous System level (AS), from the lower to the upper.

The routing in NetInf can be described in two processes: the content registration and the content discovery. If a content provider wishes to register a content, then at first it will map the content to an access node by the AN level DHT algorithm. The access node contain two information bases:1) it can identify which content can be found at which host 2)it can memorize the address or any other access information for reaching which host.

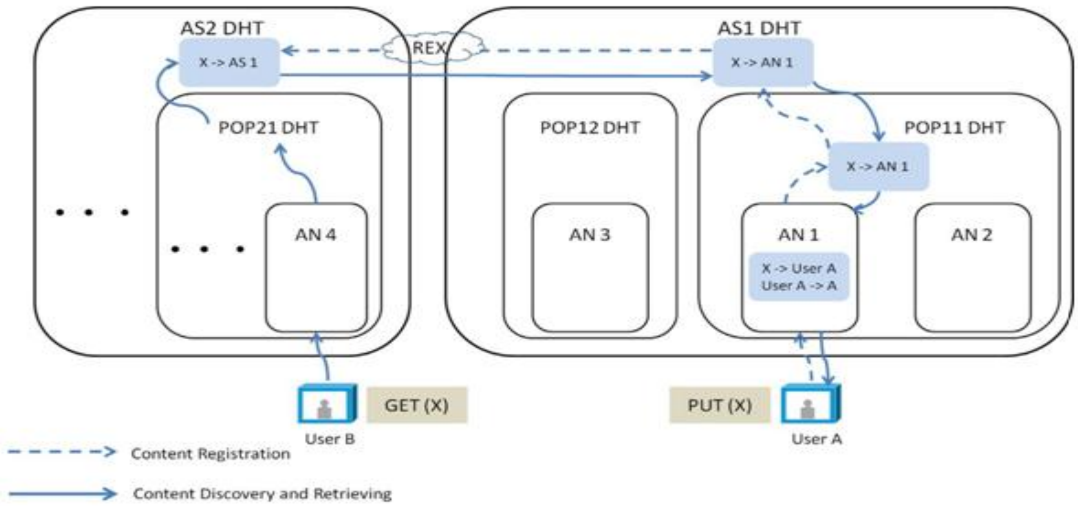

Figure 4. The routing scheme of NetInf.

3. Data oriented network architecture (DONA)

The DONA (Koponen et al., 2007) illustrates a clean-slate ICN network. The basics of the routing in DONA are a hierarchical naming resolution system with a flat content naming. The naming issue in DONA has the same characterics as that of NetInf. A flat P:L naming structure is also used by DONA, where, P is the hash of the public key of the content owner (the Principle concept in DONA). L is the owner assigned content label. The content owners have the liability to make sure that the entire P:L name is globally unique.

Although the naming issue is same as NetInf, the routing is different. DONA uses a hierarchical content name resolution system (Fig. 5), the Resolution Handles (RHs). Each RH node has a information base that contains three tuples, the content name P:L, the next hop and the distance. The next hop is from where the node receives the content name advertisement. The DONA routing contains content FIND and contains REGISTER two processes. The two processes are directly based on the flat content name. As a RH receive a REGISTER message, it will add the ⟨P:L, next hop, distance⟩ into its register table for a new arrival message. Or elseit will update the next and distance for an existing entry if the new arrival message has a shorter distance. After that, the RH will forward this message to its parent RH(s). Finally the content registration will end at the top root RH(s).

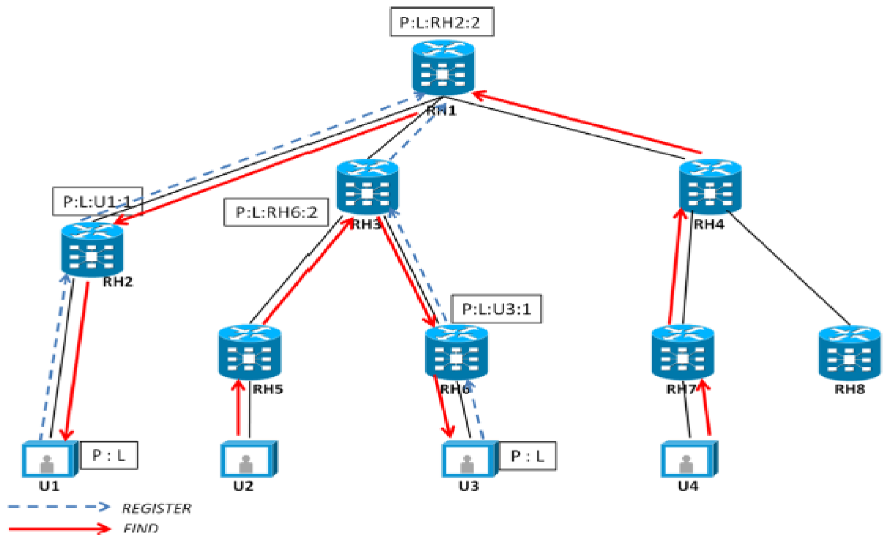

Figure 5. The routing scheme of DONA.

4. Publish subscriber Internet routing paradigm (PSIRP)

PSIRP (Lagutin et al., 2012) is an European FP7 project that initiated in 2008 and finished at 2010. PSIRP anticipated a clean slate ICN architecture that depends on a publisher-subscribe solution. PSIRP deployed the same P:L naming structure as the DONA and NetInf. The content name is referred as Resource Identifiers (RIds). The PSIRP networking follows a basicc concept known as Scopes, Scope Identifiers (SIds). The Scopes is in charge of managing and organizing the characteristics of content that includes access right, authorizations, availability, reachability, replication, persistence and the upstream resources. T he content publication (publish) and the content request (subscribe) of a content are based on a pair of composition of ⟨SId, RId⟩.

There are four significant parameters in PSIRP routing scheme.They are, RendezVous Nodes (RN), Topology Nodes (TN), Branching Nodes (BN) and Forwarding Nodes (FN). The whole PSIRP networking is divided into Domains, which is identical to the Autonomous System of the present Internet (Fig. 6). Each domain contains one RN, one TN, one BN and several FN. The purpose of RN of each domain is to match content publishers and subscribers, locating of the content publications and the scopes. Each RN can have its own name resolution system. All the RNs of every domain are interconnected with DHT into a global RendezVous Interconnection (RI) allowing the scopes of each domain to be globally reachable. The TN manages the intra-domain networking topology and load balancing. Along with that i t also exchange the path vector information with the other inter-domain TNs. The BN develops a routing map for routing the subscriber interests toward the inter-domain or intra-domain content containers by using the topology which is maintained by the TN. Lastly the role of the FNs is to utilize a Bloom filter based forwarding implementation to comprehend the content forwarding from the content container to the subscribers. The Bloom filter which is named Forwarding Identifier (FId) is collected during the subscription delivery.

In conclusion, in PSIRP a subscriber delivers its subscription of a content to the local RN of its domain so that it can obtain the content container location. The BN employ the networking topology which is acquired by the TN to forward the subscription to the content container. The subscription packet accumulates the return path into the Bloom filter and constructs the FId. At last the FNs use the FId to return the required content to the subscriber.

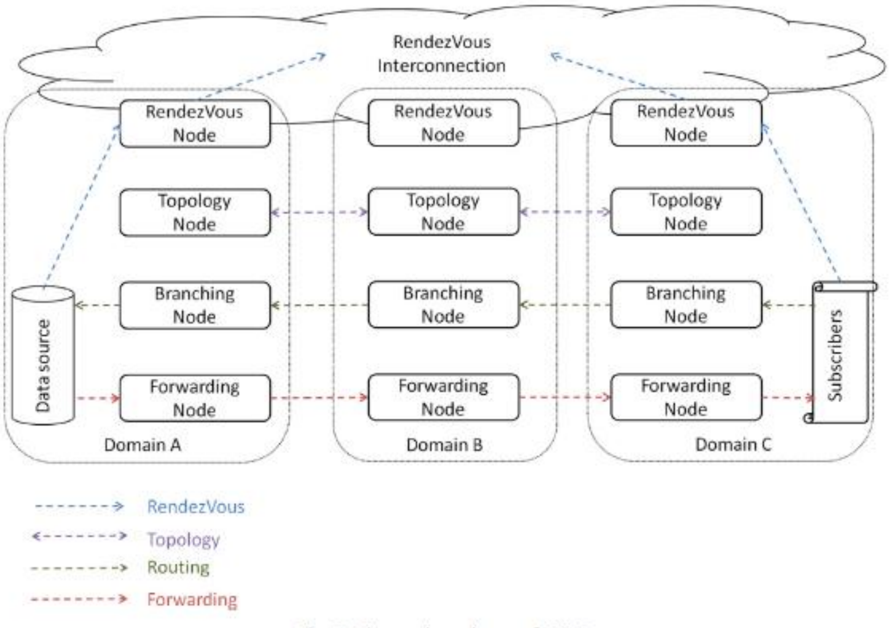

Figure 6. The routing scheme of PSIRP.

1.3.2 CACHING STRATEGIES IN ICN

The c aching methods in ICN [21] depends on their deployment form, managing algorithm, placement and replacement. In this study we shall be covering the caching management techniques. Recent literatures [21] [22] [23] [24] has shown that the LCD, LCE and ProbCache received more recognition among other cache management techniques.

1. Leave Copy Everywhere (LCE)

LCE is a process of cache managementt which is used in the traditional web management and deployment of the current Internet [25]. In LCE, once a network N exists in an ordered hierarchy and a request is placed by a subscriber (User).

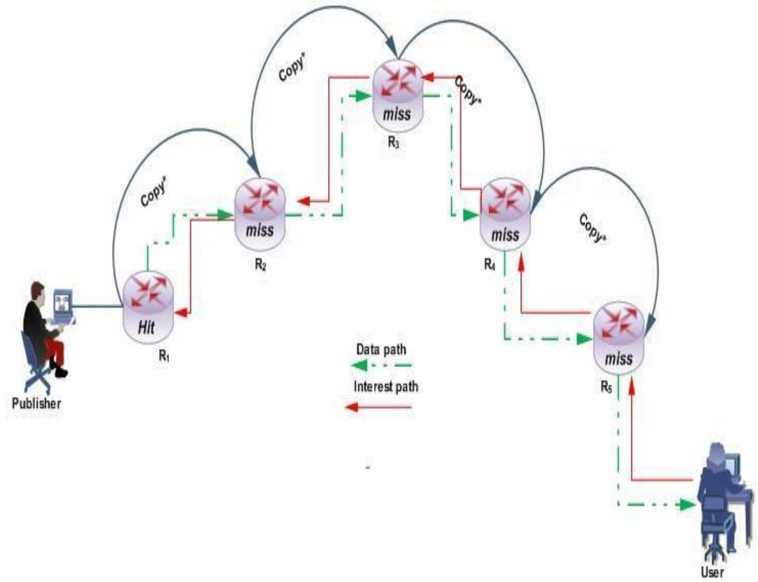

Figure 7. Leave copy everywhere.

As the request arrives, it is publicized or broadcasted in the network by using the routing and forwarding algorithm in the action. In Figure 7, the Publisher spreads out the requested-data by means of channels and nodes serving as its neighbor. In LCE, the request will keep crossing nodes until the cached data is found. The process of looking for the desired data and not finding it is called *cache-miss*. In Figure-2, it is seen that the router with named R_1 holds the desired data. By using LCE method, copies chunk of the data into all routers, on-path as a replica. In other words, the wanted data is cached in all routers on the path from the content sources to destination. Although this method improves cache-hit, but it causes accumulating redundant data, waste of bandwidth and path delay.

2. Leave Copy Down (LCD)

In LCD, the caching strategy is different. Here, LCD method focuses on avoiding the number of cache redundancy but suffers from hit network *stress*. The term stress can be defined as the distance that a node has to to cover before caching a data or communicating with another router node. In LCD, a cache-hit router (cache found), drops a copy of the data chunk to its immediate neighbor node. Figure-8.; below represents the operation and cache manageability in LCD. Here, the router R_5 - R_1 shows the cache relationship between the content requester and the publisher.

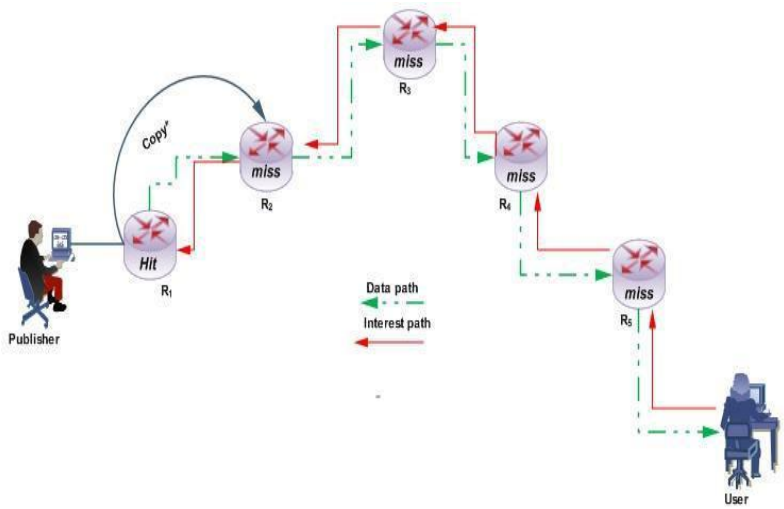

Figure 8. Leave copy down.

3. Move Copy Down (MCD)

The method of Move Copy Down (MCD) focuses mainly to minimize the amount of replica data chunk on nodes. Nonetheless, MCD is well accepted for its fast option in data accessing. This is due to its function of uniquely deleting the copied data from a local repository. When a request arrives at the network, initially a path is searched throughout the nodes to determine the cached copy. When a node is visited and the data is not found, a miss is recorded whereas a cache hit returns when the content is found. All the aforementioned cache managements, it is seen that every previous node knows the information of its neighbor. Node cognizant played the role that enables the MCD, LCD and LCE engage in the copying. The main limitation of MCD turns out to be, when several requests are placed just immediately in a neighbor node bearing little stress to the local replica. MCD therefore deletes the copy in the local replica without interfering with the main origin. Figure-9 clarifies the relationship in MCD and *cache-hits*.

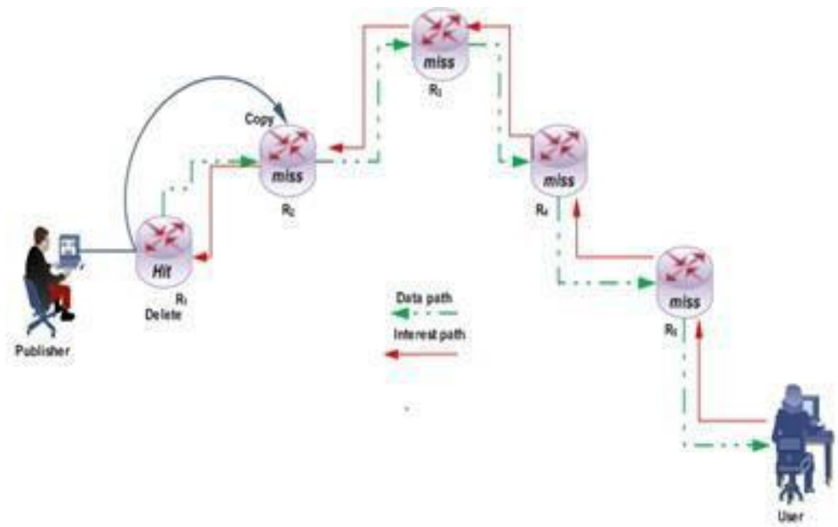

Figure 9. Move copy down.

4. Probabilistic Cache (ProbCache)

The caching strategy named ProbCache is an extraordinary and efficient cache management practice. This is because it is not aware of its previous node. Generally, it is assumed to be an extended version of LCD. Similar to other techniques, in ProbCache, routers request for data. If the data is not found, it is termed as *Miss, as* defined in LCD, LCDE and MCD. In case the data is found, the cache node copy of the data chunk is moved and copied into a successive node on-path along with its distinction in placement using the suitable probability estimation. The nodes are copied in an (*l-1*) manner as described by Loutaras *et al*., and Psaras *et al.*, [25] [26]. The method is described in Figure 5, here the cache operation is places a copy with the probability *p* and thus not satisfying the (*1-p*) *q* node. This is assumed to be fair as compared to the LCD, and LCE. In ProbCache, as the node is probabilistically selected, it may consecutively become a *proxy* to the user. A good cache-hit is always feasable when nodes have to pass through fewer routers with improved stress to a content-cache. In Figure 10, the ProbCopy assumes an immediate hit only for diagram illustration

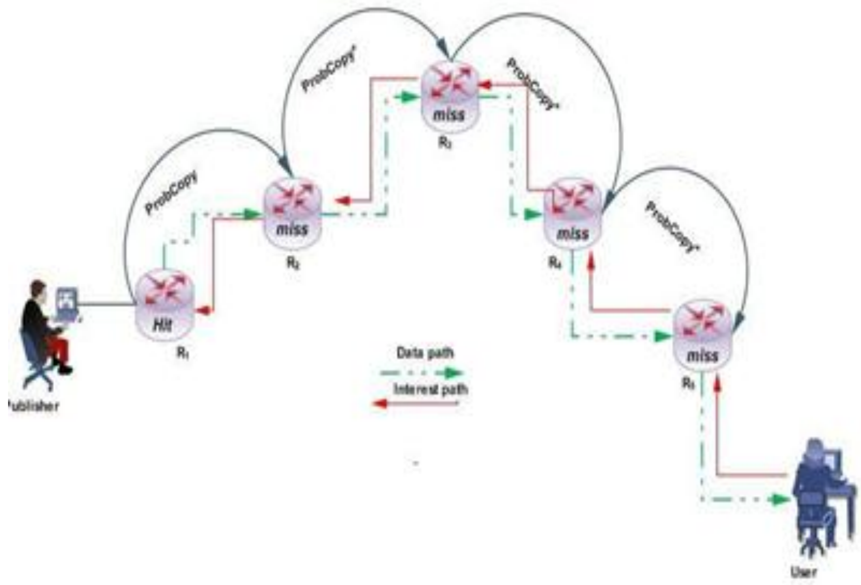

Figure 10. ProbCache.

1.3.3 Advantages Of ICN Approach

One of the major advantages of ICN [18] is the efficient content distribution. However, according to recent argument [27], this is not sufficient to stimulate a switch to a new infrastructure. The following section describes some other advantages that support the motivation.

1. Scalable and Cost-Efficient Content Distribution

The global IP traffic is predicted to rise by a factor of four from 2010 to 2015 that might reach 80 exabytes/mo in 2015. Particularly, the global mobile data traffic is likely to enhance 26 times in the time period of 2010 and 2015. The traffic contributes to various forms of video (TV, video on demand [VoD], Internet video, and P2P) that will continue to increase to a value of 90 percent of global consumer traffic by 2015.

The high demand for mass distribution and copying of large number of resources causes two main developments: P2P networking and content distribution networks (CDNs). These methods tend to move towards a more content-based communication model: uniform resource identifiers (URIs) and DNS names are defined such that, they allow accessing cached copies of content in the network.

The main challenge to produce a scalable and efficient content distribution is, as the clients and client agents are more concerned in accessing named content without having any interest of endpoint locators, there should be a way that is more architecturally sound to address such requirements. So that, there will be no need of does not require individual adjustment for specific domains and architectures. ICN is willing to provide an answer.

2. Persistent and Unique Naming

The characteristics of today's existing network are to search for objects that show IP address of a web server after DNS resolution. It serves requests by resolving the local part of URI. This causes the name-object binding to devastate easily. For example, when an object is moved from one location to another, the site changes domain, or the site for some reason is unreachable. Furthermore, if replicas of the same object are copied at different web servers, they will be available using different URIs, and in effect appear as different objects to the system (including caches).

The ICN technology diminishes these problems by using persistent and unique naming of NDOs, and with its service models those separates producers from consumers.

3. Security Model

The existing network security takes care of the communication channel between a client and a server using Transport Layer Security (TLS) or a similar technique. In this security model, the client should trust the server to deliver correct information through the channel. On the contrary, the ICN security model provides name-data integrity and origin verification of NDOs, independent of the immediate source. The model assures ubiquitous caching along with name-data integrity and authenticity.

4. Mobility and Multihoming

The current network architecture causes complication of end-to-end connection in terms of mobility and multihoming of nodes. Also with choosing which route or interface to use for these connections.

ICN approach eliminates end-to-end connection. The problem becomes less complicated. A moving client continues to issue requests for NDOs on a new access. Requests on the new access are potentially served from a different source, instead of needing to maintain a connection to the previous source. A multi-homed client can similarly choose to send a request on any one, several, or all accesses.

5. Disruption Tolerance

To achieve end-to-end communication with transport sessions to origin servers is often difficult to achieve in challenged networks, with sparse connectivity, high-speed mobility, and disruptions. As application protocol sessions are bound to transport sessions, they will fail as soon as the transport session fails.

The seamless communication with end-to-end paths [28] is not required by several applications. If the main purpose is to access data objects, ICN can utilize its convergence layer idea for hop-by –hop transport, within its in-network caching which can offer store-and-forward approaches similar to the data transport networking (DTN) architecture [29]. This will ensure enhanced reliability and performance by leveraging optimized hop-by-hop transport and in-network caching.

Chapter 2

Architecture and model

Chapter 2 Architecture and model

2.1 Node Architecture

In named data network (NDN), on-path caching approach is made. The work of NDN includes content centric networking (CCN) proposal. We deploy the CCN node architecture and naming concept and the forwarding engine model. According to the perspective of CCN node, the number of requests and responses should be in equilibrium. This means for each request, there will be one response (or no response). Based on local configuration, observed network performance, and other factors, CCN nodes can use various *strategies* for requests (re-) transmission pace and interface selection [18]. The first thing that NDN router does after an Interest for content hits is, it searches for the content object in the content store (CS). In this scenario two cases can occur, the object will be found or not. In case of former, NDN will deliver the requested content to the requester. In case of latter, two tables i.e. Pending Interest Table (PIT) and Forwarding Information Base (FIB) are used to manage the Interest packet. PIT accumulates incoming interfaces of the Interest packets so that the data .PIT to determine whether the same content was previously asked but is still unanswered. If such a content entry is found in PIT, and then the incoming interest interface is noted so that when the cached data arrives, all the requests are met. Whereas if the entry is not found, the new Interest is forwarded to FIB and FIB directs Interest packet towards one or more content sources/ repositories. [30]

2.2 Naming Model

CCN uses name-based routing. As a client seeks for an object by sending interest packets, it is routed toward the publisher of the name prefix via longest-prefix matching in the forwarding information base (FIB) of each node. The FIB can be constructed with routing protocols similar to those of existing Internet. The CCN nodes keep state for each outstanding request in the pending interest table. As a result request aggregation is possible, i.e., when multiple requests arrive on the same node for the same NDO, only the first is forwarded towards the source. When the copy of the desired data is found, it comes back through the reverse path to the client (all nodes along the path cache a copy of the object). The reverse path is found using the state that the interest packet has left in the nodes [18].

Firstly, we deploy the CCN naming concept. Along with that, we introduce naming of every Router with in an AS. The name of each router is unique within an AS. (The name is unique within an AS.) An identical name can be given for the routers if they belong to different AS. To avoid complexity, we use alphabetic naming in our paper. Although keeping meaningful name (scalable, unique, and easy to remember) will also be a good practice. In the second step, we leverage the architecture of the CCN node. According to our model each node consists of three main data structures namely the CS, the PIT and the FIB. [31]

Similar to NDN, the function of FIB is to direct Interest packets towards potential sources (permanent copy) of the same data. The CS will have two tables: Data Table and Cache-Route Table. The function of data table is to collect data and that of Cache-Route Table stores the path towards another node (closer router) which might contain the temporary cache copy of the desired content. Content Store's Data Table match will be chosen over PIT match, Cache-Route Table match of Content Store's and FIB match. PIT match will be preferred over Content Store's Cache-Route Table match and FIB match. Among Content Store's Cache-Route Table match and FIB match, the selection will be inclined towards Cache-Route Table match.

Chapter 3

Proposed Forwarding and Caching Strategy

Chapter 3 Proposed Forwarding and Caching Strategy

3.1 Forward Engine

As an Interest or request for a content arrives at the CCN router (router with leveraged architecture, proposed in Section 3) , firstly CCN searches for the content in the Data Table of its own CS. If the desired data packet is already in the Data Table of the CS then the matching data packet goes back to the original requester by following the trace of ‘bread crumbs’ left by the Interest packet. Immediately the Interest will be removed (as the request is already satisfied). In case, when the requested data packet is not found in the Data Table of the CS, PIT will be used to deal with the Interest packet. The purpose of PIT is to map information object names to the requesting face (or faces) of the content and it aggregates Interests and multicasts data to the requesting users as defined in NDN [2].

On the contrary, if the PIT entry for the corresponding Interest is not found, then the following phase comes to handle such condition. In ICN the data delivery process activates by user requests which are forwarded towards a copy of the requested item. Forward engines (e.g. FIB) are responsible for request forwarding which contains information about reachability of different content items. With the existence of a highly distributed caching infrastructure, it is seen that due to spread-out of short-term replicas throughout the network, item accessibility and location can vary with time. There are a number of factors that affects replica distribution as content popularity or caching policy, and the request forwarding scheme should be modified accordingly. High level view of forwarding engine of our proposed scheme is shown in Figure 1.

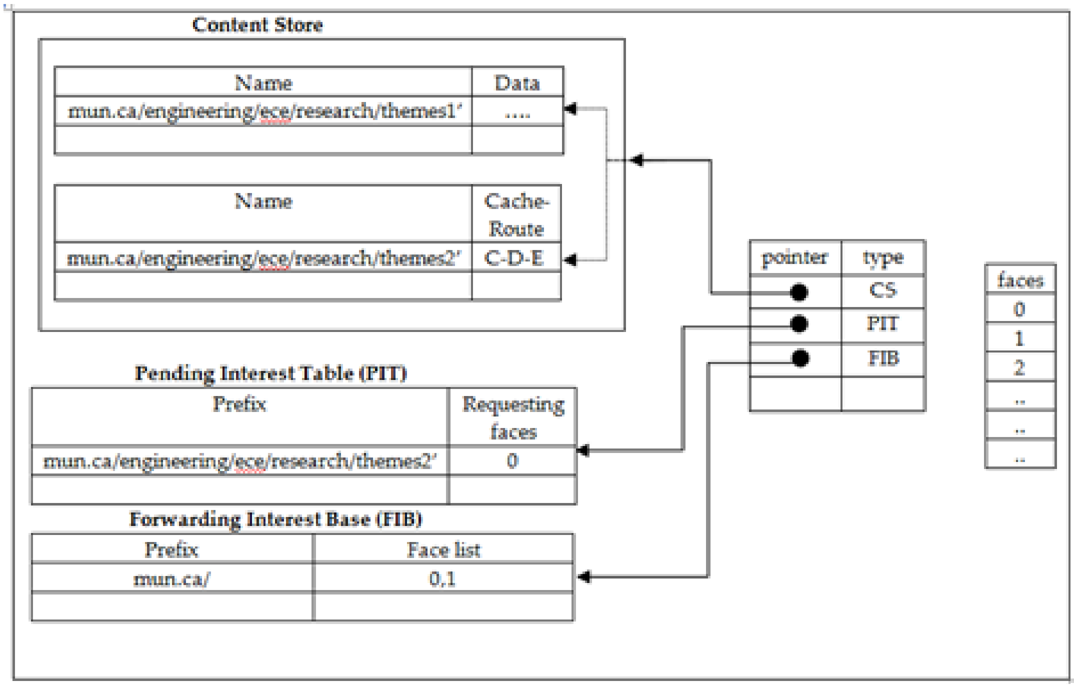

Figure 11. High level view of forwarding engine of Cache-Route strategy.

3.1.1 Exploiting Phase

The content is looked into the PIT, if not found, the router searches for the content name at the Cache-Route Table. If information of path towards the temporary copy is found in the Cache-Route Table the process enters into the *Exploring Phase.* OR else, forwarding face is elected from the FIB. The purpose of FIB is to store the mapping of aggregated name prefixes to the next hop forwarding face(s) towards the information publishers. The FIB forwards Interests from the consumers towards longest prefix matching publishers/repository. This forwarding mechanism (*Exploiting Phase*) will persist until it finds a match for the data (either Content Store's Data Table match or PIT match or access the repository) else it enters to an *Exploring Phase*.

3.1.2 Exploring Phase

The node will begin the *Exploring Phase* if the corresponding route/path is found in the Cache-Route Table of CS. As the Interest proceeds to explore the possible cache-copy, it will follow the forwarding engine mechanism, described below. A PIT entry will be given and the Interest will be forward with the cache-route. In *Exploring Phase* the Interest will go through the route as suggested at the *Exploring Phase* initiating node. Only the function of Data Table match of CS and matching data sending back through the breadcrumbs of the Interest route, will work during the exploring phase. Other activities of PIT and FIB associated to that Interest will be excluded for the time being. If the desired data is found it will be sent back to the *Exploring Phase* initiating node and subsequently it will be reached to the requesting nodes as well. The node will enter into the *Exploiting Phase* to forward the Interest to the permanent copy if the content is not served to the node generating *Exploring Phase* within a given time limit.

Routers do not possess the information of address of other routers except their 1-hop neighbors. Thus in the *Exploring Phase* where router forward Interest to the next hop, it is required to provide the knowledge of route or path. Hence by analyzing the route table, router can only forward the Interest to the next hop. In this way, the Interest is indicated towards the path to the possible cache copy of the desired content.

It is important to notice that for every request, *Exploring Phase* will be implemented maximum one time. If the request cannot be achieved through the *Exploring Phase* and the time limit crosses, it will be received through the *Exploiting Phase.* This provides an advantage of loop-free executions of our proposed scheme.

3.2 Caching Policy

According to our On-path caching strategy, copying or caching is done on one router abiding on the route between publisher (local repository and subscriber (client). Some other routers along the path stores the route where the content is cached, mapped with the corresponding content name, given that the distance(hop counts) of the cache copy should be less than the distance(hop counts) of the permanent copy. When the data is re-captured from the local domain (Inter-Domain traffic), the data is cached only on one router between the Gateway node to the generating node (Interest originator).

We deploy last node (Interest generator node) caching strategy. In this method, after the content is received from the repository (permanent copy), it is cached in the requesting node. Other routers in the path (routers that figure out that the cached copy is not farther than the permanent copy) will upgrade their Cache-Route Table entry (with timeout) of the CS for the desired content. To summarize, it is seen that one router will cache the object along the path and other routers will only contain the information of route by which it can re-direct an Interest towards the desired cached copy.

Also when object is collected from the cache/copy, no more replicas are being made. The strategy described above is explained with the Fig. 1-5 and also pseudo code of the algorithm is given in Table 2.

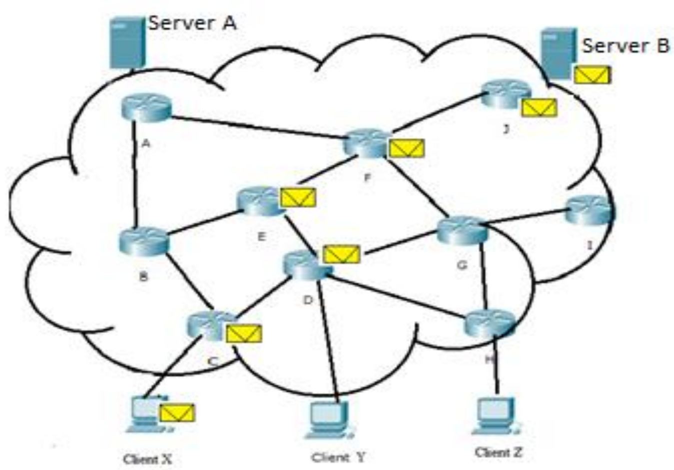

Figure 12: Local Connectivity of AS's Routers

In Figure 1, let, a repository adjacent to A claims that it can provide Interests matching the prefix 'mun.ca/engineering/ece/research/'. The router A broadcasts the prefix 'mun.ca/engineering/ece/research/' in prefix announcements using CCN TLV [32]. Let an Interest is requested by a client (*client X*) adjacent to C for the object in 'mun.ca/engineering/ece/research/themes'. Suppose that the Interest will be forwarded to A via the path C-D-E-F-J and J will direct it to its adjacent repository. Here, similar to NDN the Interest packet generates a trail of 'bread crumbs' for a matching content packet to go back to the original requester.

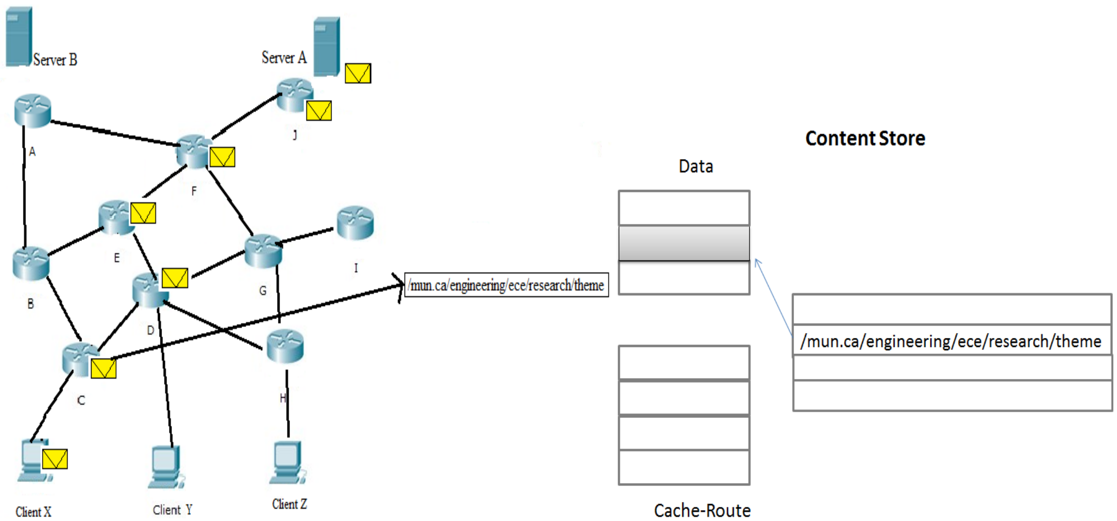

Figure 13: Content Store of Router C after retrieving the content 'mun.ca/engineering/ece/research/themes' from the repository next to J

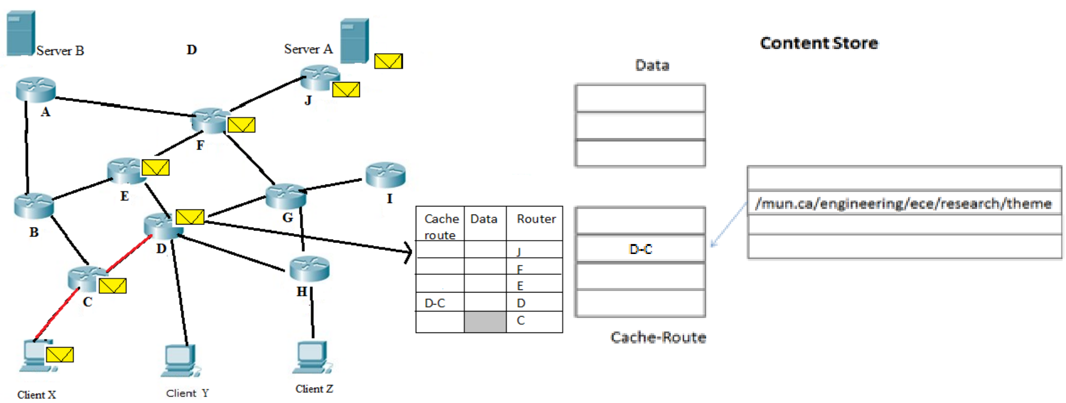

Figure 13: Content Store of Router D after retrieving the content 'mun.ca/engineering/ece/research/themes' from the repository next to J

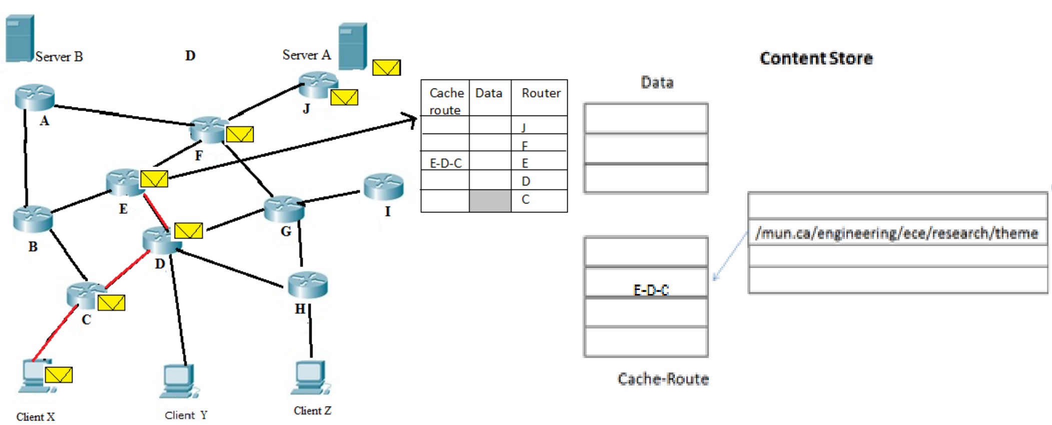

Figure 14: Content Store of Router E after retrieving the content 'mun.ca/engineering/ece/research/themes' from repository next to J

Cache-Route	Data	Router
		J
		F
E-D-C		E
D-C		D
		C

Figure 15: Content Store for the content 'mun.ca/engineering/ece/research/themes' of the path J-F-E-D-C after retrieving it from repository (next to J) by *client X*

Let the same content '/mun.ca/engineering/ece/research/themes' is requested by the *client X*. The object will be provided through the optimum path J-F-E-D-C . According to our strategy, the content will be cached on only one router from data source (repository next to J) to requesting node (C). While some other routers along path will cache only the route of the cached data depending on the hop-count of the permanent copy (information stored at FIB).

Based on the last node (Interest generator node) caching policy, the data 'mun.ca/engineering/ece/research/themes' will be cached only on one router (C) (shown in Figure 2 and Figure 5).Whereas, the routers E and D will cache only the route towards the router C (mapped with 'mun.ca/engineering/ece/research/themes' in Cache-Route Table) rather than caching the initial data (shown in Figure 3-5). The routers E and D will cache the route towards the router C (which will hold the cache copy) as the hop distance of C from E and D are not larger than the repository distance fromE and D . In the same way, the route will not be cached on routers J and F towards the router C (which holds the cache copy) since the hop distance of C is not larger than the repository distance from J and F .

Let an identical Interest in '/mun.ca/engineering/ece/research/themes' is requested by another client (*client Z*) residing next to H and the Interest is likely to be forwarded to repository next to J . Let us consider that the shortest part is H-D-E-F-J (Figure 2). As the Interest arrives on router D ; D will initiate the *Exploring Phase* and send the Interest to C as D has the information that the content '/mun.ca/engineering/ece/research/themes' is cached by C . Thus, the content will be served by C (temporary cached copy) to the H along the path C-D-H . Due to the fact that the object is collected from the cache, the requesting router (H in this case) will not make any further replica of the object in its Data Table of CS. Hence caching is done only if the data is retrieved from a permanent copy (or repository) as a result minimizing the number of replica.

Thus, even if the router containing the cached copy of the content is not en-route of the requesting node (client) to the repository; our proposed scheme can still identify and serve the content from the cache; as long as an en-route router has the information of the route towards the router with the possible cache copy of the content. Further to this, since the content is cached only one router during one fetch, it leaves room for the other content resulting in accommodating more varying contents in the locality. As number of diverse contents reside in the nearby cache (locality) is increased, and Cache-Route Table helps to locate the cache copy, so the probability of finding an independent requesting content in the locality rises. This causes a depletion of server/ repository hit ratio.

3.2.1 Caching Algorithm

For cache eviction algorithm, we use Least Recently Used (LRU) replacement policy (both for Data Table and Cache-Route Table of CS). This is shown in Table 2.

Pseudo-code for forwarding Interest towards data Source(upstream)	***Pseudo-code for forwarding Data from source to requester (downstream)***
i:Node where Interest is generated R: The repository that contains the requested data CRT: Cache-Route table j:the node attached/next to R CR: the nodes/ routers in the network 1. Initialize (CR ←i) 2. Initialize (Explored ← 0) 3. WHILE (CR != j) 4.<u>begin</u> 5. IF (data is found in Data table of CS) 6. return CR <u>7. END IF</u> <u>8.</u>ELSEIF(Explored!= 1) and (data temporary 9. location,T is found in CRT of CS) 10. WHILE (CR != T) 11. <u>begin</u> 12. Explored ← 1	1. C: Source node (Cache/temporary copy) of the 2. requested data 3. R: Source node (Repository) of the requested data 4. i:Node where Interest is generated 5. S ← CR value retrieved from *upstream* Algorithm 6. While(CR != i) 7. <u>begin</u> 8. CR ← Next hop towards i 9.forward Data to CR 10. IFhop_count (R) ≥hop_count (C) 11. Update CRT of CS//Update cache 12. knowledge 13. <u>end</u> 14. IF (S = = R) Store data at CS(Data table) of i 15.END IF 16. IF (S = = C) Update CRT of CS 17.END IF

```
        CR ← Next hop towards T
        forward Interest to CR
    IF (data is found in Data table of CS)
        return CR
        END IF
          end
   CR  ← Next hop towards j
   forward Interest to CR
end
```

Table 2: Pseudo code representing cache eviction algorithm

Chapter 4

Results and Discussion

Chapter 4 Results and Discussion

4.1 Numerical Analysis

4.1.1 Load Balancing Ratio

i) Load Imbalance: We quantify load imbalance as the ratio between the maximum load of the cache (cache containing most popular contents hence serves highest traffic) and the average load of a cache. We further assume that each cache corresponds to a router within the AS. After generating request for an object the request is randomly distributed to a client attached to any of the router. Thus the content requests follow the Independent Reference Model [33].

Let L is the random variable corresponding to the fraction of request served by node N_i received by the clients connected to nodeN_i .Then the load imbalance, represented by the co-efficient of variation of a random variable, is the ratio between its standard deviation and its mean value. From [34] [35], we can simplify the load imbalance,$C_v[L]$ as follows:

$$C_v[L] = \sqrt{N-1}\sqrt{\sum_{i=1}^{F} p_i^2}. \quad (1)$$

where*N* is the total number of nodes, and *F*is the number of objects.

$$p(i) = \frac{1/i^{\alpha}}{H_n^{(\alpha)}}; \quad (2)$$

$$H_n^{(\alpha)} = \sum_{j=1}^{F} 1/j^{\alpha}; \quad (3)$$

where $p(i)$,expresses the content popularity for object i, which follows a Zipf distribution [36], with exponent α and $H_n^{(\alpha)}$is the rank of the i-th most popularfile in a catalog of size F files.$H_n^{(\alpha)}$can be approximated as follows,

$$H_n^{(\alpha)} \approx \int_1^{F+1} \frac{dx}{x^\alpha} = \begin{cases} \frac{(F+1)^{1-\alpha}-1}{1-\alpha}, & \alpha \neq 1 \\ \log(N+1), & \alpha = 1 \end{cases} \quad (4)$$

Let, h_i is the hit probability of item i, that can be found by the client at its attached node, From equation (1) we get,

$$C_v[L] = \sqrt{N-1}\frac{\sqrt{\sum_{i=1}^{F} p_i^2(1-h_i)}}{\sum_{i=1}^{F} p_i(1-h_i)}. \quad (5)$$

Let, r is the r-th most popular object in the catalog that is cached in a node. For simplicity, if we consider, in a certain interval the node will have the cached content that are most popular for the client attached to it i.e., the node cached the content of decreasing popularity of rank 1,2,……. r, then we get,

$$C_v[L'] = \sqrt{N-1}\frac{\sqrt{\sum_{i=1}^{r} p_i^2}}{\sum_{i=1}^{r} p_i}. \quad (6)$$

From equation (4) and (6) we get,

$$C_v[L'] \approx \sqrt{N-1}\frac{\sqrt{\frac{(r+1)^{1-2\alpha}-1}{1-2\alpha}}}{\frac{(r+1)^{1-\alpha}-1}{1-\alpha}}. \qquad (7)$$

4.1.2 Cache Hit Ratio

A node having a particular content can receive requests for that content in two ways. Firstly, request is issued by a client who is attached to the node which is capable of serving the request. Secondly, the request is forwarded from other node of the network. In the later case, request comes through Exploiting Phase (path getting from FIB) or Exploring Phase (path getting from Cache-Route table).

Letr_{ij} be the incoming rate of requests for object$F_i$at cache n_j ,λ_{ij} is the rate of request for F_i issued by the client connected to cache n_j and let m_{ih} be the rate of requests for F_i in the miss stream of node n_h (*Exploiting Phase*) andxl_{ih} is the cache route stream from other nodes(*Exploring Phase*). Then, for all $F_i \epsilon F$ and all$n_j \epsilon N$. Therefore,

$$r_{ij} = \lambda_{ij} + \sum_{h:R(n_h,\ F_i)=n_j} m_{ih} + \sum_{h:R(n_h,\ F_i)=n_j} xl_{ih}, \qquad (8)$$

The cache hit ratio $\pi_i$can be approximated based on the work [37] where value of r_{ij} can be found from the equation (8). Therefore,

$$\pi_i = 1 - e^{-p_i r_{ij}}. \qquad (9)$$

4.2 Simulation Results

Simulation is done with ccnSim [38] which is a scalable chunk-level simulator of CCN under the OMNeT++ framework[39]. The network topology used in the simulation is illustrated in Fig. 1. The figure shows 10 CCN nodes and each node has a cache size of 100 objects. Two repositories are shown in the diagram connected with the node A and the node J. A number of 10,000 objects are replicated/ stored on both the repository next to A and the repository next to J.   Each node ends with a client connected to it. The aggregate request stream for an object i arrives to each CCN node with a rate, $\lambda_i = p(i)\lambda$. Request generation rate follows Poisson distribution and randomly tagged with one of the 10 CCN nodes given that for each CCN node the aggregate request rate over all objects is $\lambda \leq 1$ Hz. Cache object displacement policy used is LRU. Cached content's route (eviction at Cache-Route Table) eviction policy is also LRU.

For LCE, we suppose that the shortest route repository from FIB is chosen by the strategy layer. In addition, the CS and the PIT behave as proposed in NDN [2]. In our proposed Cache-Route scheme, the strategy layer also chooses the shortest path repository. Nevertheless, at each hop, based on Cache-Route Table status of Content Source (CS) forwarding path engineering is made. The simulation is produced for the following performance metrics:

4.2.1 Average Hop Ratio

(1) Average Hop-distance: One of the superior goals of all networks is to reduce the average hop-distance. In the perspective of user, decreasing the hop-distance refers to low end-user latency (low round trip delay). On the other hand, from the perception of network, lower hop distance means lower traffic flow/network load.

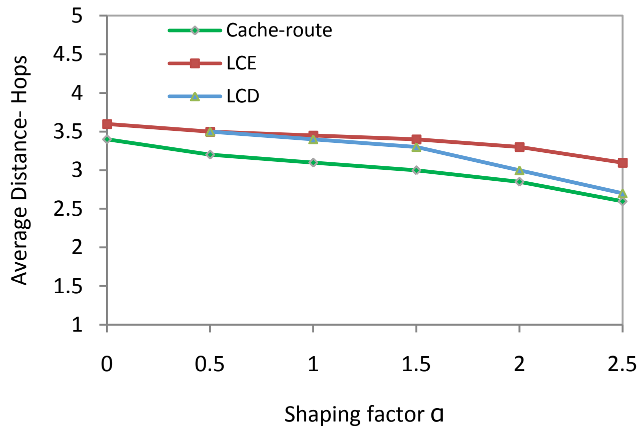

Figure 16: Comparison of Average hop-distance for Cache-Route strategy, LCE policy and LCD policy.

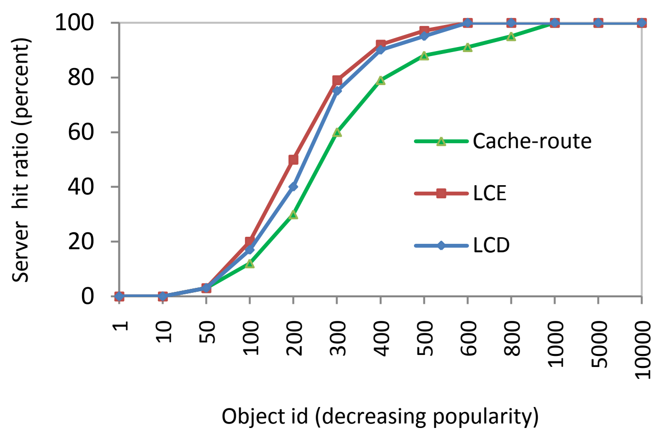

Figure 17: Comparison of Server/repository hit-ratio between Cache-Route strategy, LCE policy and LCD policy for the objects of different popularity

Figure 16. shows a differentiation of our proposed Cache-Route scheme with the LCE scheme. It is apparent from the graph that our projected scheme outperforms both LCE and LCD policies. Note that, for small values of α, the performance difference among the schemes are low, whereas, when the value of α is increased, the Cache-Route scheme accomplish higher gain compared to LCE. LCD also performs better than LCE because LCD preserves less number of replica compared to LCE.

4.2.2 Server Hit Ratio

(2) Reduction of the server hit ratio is another important objective of deploying widespread cache structure of ICN. In Figure 17, server hit-ratio for objects of different popularity is

depicted for Cache-Route strategy, LCD policy and LCE policy. The pattern clearly shows the performance of our suggested scheme is better than the LCD and LCE schemes. It is due to the fact that, our proposed strategy affirms to cache the data in the path only if it is found in the repository. If the object is found in the cache, it collects the data from that cache and updates the Cache-Route Table entry of the CS without making any further replica of the same content on the path. This process confirms fewer replicas within the local network (intra-domain). In other words, there is a chance of accumulation of wider range of popular contents in the local network. On the contrary, the novel concept of Cache-Route/cache-path caching assists to engineer more dynamic forwarding strategy to find the temporary copy (cached data) resulting in a reduction of server/repository hit ratio.

Chapter 5

Conclusion and Future Work

Chapter 5 Conclusion and Future Work

5.1 Conclusion

In this paper, we take a holistic approach to re-examine and concentrate on the on-path caching and forwarding strategy for ICN. This is to get rid of the inefficiencies that arise from caching object in all routers (LCE) which leads to too many replicas of the same content and a shortage in cache memory for new content. We propose a novel concept of content's cache-route/path caching. We deploy this cache route knowledge in forwarding path engineering at every hop. NDN node architecture and forwarding strategy is adopted in the process. We eradicate cache redundancy, simultaneously we put purposeful effort to locate and retrieve cache copy of content rather than accessing the remote server/repository. To determine the load balance and cache hit ratio, mathematical analysis is deployed while simulation is performed to show the nature of average hop-distance and server hit ratio. The simulation and mathematical solutions support our theoretical idea and verify the efficiency of our proposed scheme.

5.2 Future Work

As a scope of future work, the comprehension of the scheme is needed in practical environment. Specifically, the effect of our proposed scheme to decrease cross operator traffic (Cross-AS Traffic) needs to be carried out in real life scenario.

References

[1] Lagutin, Dmitrij, Kari Visala, and Sasu Tarkoma. "Publish/Subscribe for Internet: PSIRP Perspective." *Future internet assembly* 84 (2010).

[2]Jacobson, Van, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs,and Rebecca L. Braynard. "Networking named content." In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pp. 1-12. ACM, 2009.

[3]Lopez, Jairo E., Mohammad Arifuzzaman, Li Zhu, Zheng Wen, and Sato Takuro. "Seamless mobility in data aware networking." In *ITU Kaleidoscope: Trust in the Information Society (K-2015), 2015*, pp. 1-7. IEEE, 2015.

[4]Agyapong, Patrick Kwadwo, and Marvin Sirbu. "Economic incentives in information-centric networking: Implications for protocol design and public policy." *IEEE Communications Magazine* 50, no. 12 (2012).

[5]Arifuzzaman, M., K. Yu, and T. Sato. "Collaboration between network players of information centric network: An engineering-economic analysis." *Journal of ICT Standardization* 2, no. 3 (2015): 201-222.

[6] Mathieu, Bertrand, Patrick Truong, Wei You, and Jean-François Peltier. "Information-centric networking: a natural design for social network applications." *IEEE Communications Magazine* 50, no. 7 (2012).

[7]Yu, Keping, Mohammad Arifuzzaman, Zheng Wen, Di Zhang, and Takuro Sato. "A key management scheme for secure communications of information centric advanced metering infrastructure in smart grid." *IEEE transactions on instrumentation and measurement* 64, no. 8 (2015): 2072-2085.

[8] Nguyen, Quang N., Mohammad Arifuzzaman, and Takuro Sato. "Proactive-caching based information centric networking architecture for reliable green communication in intelligent transport system." In *ITU Kaleidoscope: Trust in the Information Society (K-2015), 2015*, pp. 1-7. IEEE, 2015.

[9] Tang, Xueyan, and Samuel T. Chanson. "Coordinated en-route web caching." *IEEE Transactions on Computers* 51, no. 6 (2002): 595-607.

[10]Laoutaris, Nikolaos, Hao Che, and Ioannis Stavrakakis. "The LCD interconnection of LRU caches and its analysis." *Performance Evaluation* 63, no. 7 (2006): 609-634.

[11] Psaras, Ioannis, Wei Koong Chai, and George Pavlou. "Probabilistic in-network caching for information-centric networks." In *Proceedings of the second edition of the ICN workshop on Information-centric networking*, pp. 55-60. ACM, 2012.

[12] Zhang, Meng, Hongbin Luo, and Hongke Zhang. "A survey of caching mechanisms in information-centric networking." *IEEE Communications Surveys & Tutorials* 17, no. 3 (2015): 1473-1499.

[13] Katsaros, Konstantinos, George Xylomenos, and George C. Polyzos. "MultiCache: An overlay architecture for information-centric networking." *Computer Networks* 55, no. 4 (2011): 936-947.

[14] Chiocchetti, Raffaele, Dario Rossi, Giuseppe Rossini, Giovanna Carofiglio, and Diego Perino. "Exploit the known or explore the unknown?: hamlet-like doubts in icn." In *Proceedings of the second edition of the ICN workshop on Information-centric networking*, pp. 7-12. ACM, 2012.

[15] "Network of Information (NetInf)." Internet: http://www.netinf.org/sail-netinf/, [Feb. 26, 2016].

[16] PURSUIT: PURSUING A PUB/SUB INTERNET. Available online: http://www.fp7-pursuit.eu/PursuitWeb/, (accessed on May 26, 2017).

[17] “COntent Mediator architecture for content-aware nETworks (COMET).”Internet: http://www.comet-project.org/, [Feb. 26, 2016].

[18] “THE CONVERGENCE PROJECT.” Internet:: http://www.ict-convergence.eu/, [Feb. 26, 2016].

[19] Xylomenos, George, Christopher N. Ververidis, Vasilios A. Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V. Katsaros, and George C. Polyzos. "A survey of information-centric networking research." *IEEE Communications Surveys & Tutorials* 16, no. 2 (2014): 1024-1049.

[20] Vasilakos, Athanasios V., Zhe Li, Gwendal Simon, and Wei You. "Information centric network: Research challenges and opportunities." *Journal of Network and Computer Applications* 52 (2015): 1-10.

[21] Abdullahi, Ibrahim, and Suki Arif. "CACHE-SKIP APPROACH FOR INFORMATION-CENTRIC NETWORK." (2006).

[22] Psaras, Ioannis, Wei Koong Chai, and George Pavlou. "In-network cache management and resource allocation for information-centric networks." *IEEE Transactions on Parallel and Distributed Systems* 25, no. 11 (2014): 2920-2931.

[23] Abdullahi, Ibrahim, Suhaidi Hassan, and Suki Arif. "Prospective use of bloom filter and muxing for information centric network caching." *ARPN Journal of Engineering and Applied Sciences* 10, no. 3 (2015): 1169-1177.

[24] Fang, Chao, F. Richard Yu, Tao Huang, Jiang Liu, and Yunjie Liu. "A Survey of Green Information-Centric Networking: Research Issues and Challenges." *IEEE Communications Surveys and Tutorials* 17, no. 3 (2015): 1455-1472.

[25] Laoutaris, Nikolaos, Sofia Syntila, and Ioannis Stavrakakis. "Meta algorithms for hierarchical web caches." In *Performance, Computing, and Communications, 2004 IEEE International Conference on*, pp. 445-452. IEEE, 2004.

[26] Ghodsi, Ali, Scott Shenker, Teemu Koponen, Ankit Singla, Barath Raghavan, and James Wilcox. "Information-centric networking: seeing the forest for the trees." In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, p. 1. ACM, 2011.

[27] Ott, Jörg, and Dirk Kutscher. "Why Seamless? Towards Exploiting WLAN-Based Intermittent Connectivity on the Road." In *TERENA Networking Conference*. 2004.

[28] Fall, Kevin. "A delay-tolerant network architecture for challenged internets." In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 27-34. ACM, 2003.

[29] Jacobson, Van, Marc Mosko, D. Smetters, and J. J. Garcia-Luna-Aceves. "Content-Centric Networking, Whitepaper Describing Future Assurable Global Networks." *Palo Alto Research Center, Inc* (2007): 1-9.

[30] You, Wei, Bertrand Mathieu, Patrick Truong, Jean-François Peltier, and Gwendal Simon. "Dipit: A distributed bloom-filter based pit table for ccn nodes." In *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, pp. 1-7. IEEE, 2012.

[31] Zhang, Lixia, Deborah Estrin, Jeffrey Burke, Van Jacobson, James D. Thornton, Diana K. Smetters, Beichuan Zhang, Gene Tsudik, Dan Massey, and Christos Papadopoulos. "Named data networking (ndn) project." *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*(2010).

[32] N. Shen and R. Aggarwal, "Intermediate system to intermediate system (IS-IS) extensions for advertising router information," Internet: http://tools.ietf.org/html/rfc4971,Jul. 2007, [May 26, 2017].

[33] E. G. Coffman and P. J. Denning, *Operating systems theory*, vol. 973. Prentice-Hall Englewood Cliffs, NJ, 1973.

[34] Maggs, Bruce M., and Ramesh K. Sitaraman. "Algorithmic nuggets in content delivery." *ACM SIGCOMM Computer Communication Review* 45, no. 3 (2015): 52-66.

[35] Saino, Lorenzo, Ioannis Psaras, and George Pavlou. "Understanding sharded caching systems." In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, pp. 1-9. IEEE, 2016.

[36] Breslau, Lee, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. "Web caching and Zipf-like distributions: Evidence and implications." In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1, pp. 126-134. IEEE, 1999.

[37] Laoutaris, Nikolaos. "A closed-form method for LRU replacement under generalized power-law demand." *arXiv preprint arXiv:0705.1970* (2007).

[38]"ccnSim,"Internet:http://perso.telecomparistech.fr/~drossi/index.php?n=Software.ccnSim,[Feb. 26, 2016].

[39] "OMNeT++ Discrete Event Simulator,"Internet: https://omnetpp.org/, [Feb. 26, 2016].