Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2018, 5(2): 80-90

Research Article ISSN: 2394 - 658X

Integrated Caching and Routing Strategy for Information-Centric Networks

Mohammad Arifuzzaman[1], Naheed NazneenTuli1[1], Yu Keping[2], Quang N Nguyen[2] and Sato Takuro[2]

[1]*Department of Electronics and Communications Engineering, East West University, Dhaka, Bangladesh*
[2]*School of Fundamental Science and Engineering, Waseda University, Tokyo, Japan*
nntuli09@gmail.com

ABSTRACT

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) /Name Resolution System (NRS) (in routing through name resolution) or FIB (in name-based routing) is not conscious of the cached data. Therefore, the most extensively deployed intra-domain routing protocol and their forwarding strategy is not capable of addressing all available temporary cached copies of the content. Hence, on-path caching strategy experiences two major downsides: numerous replicas of same content within AS without adding significant value in terms of cache resource utilization and an approach taken to minimize redundancy ends up with the cost of fetching more copy of similar content. 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. 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.

Keywords: Information Centric Networking, Caching, On-path Caching, Routing, Forwarding

---

INTRODUCTION

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. In [2], authors illustrate Content-Centric Networking (CCN) which 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. 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 increased in overall latency of the network. The main objective is to cache objects in one router thus leaving space for other objects or popular information to be cached while needed. Here, the main objective is to overcome the problem of numerous number of caching objects. If the object is cached in all routers, although it has an advantage of achieving the requested object easily but the memory falls short so 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 extensively depend on the spreaded cache construction. The vital purpose to use widespread cache structure is to make sure reduced latency that is the 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 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. We leverage NDN node architecture, naming concept and the forwarding engine model, which are described in Section 3. Our proposed caching and forwarding strategy is elaborated in Section 4. We present the results and discussions of our proposal in Section 5. Finally, section 6, concludes the paper with a summary and some scope of future work. There are several abbreviations throughout the paper, to enhance readability of user; a table listing full forms is added at the end of this section.

Table-1 List of Abbreviations

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

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 analysing the future Internet architecture with respect to public policy. The fact that ICN's can be used 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. 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 social networking [6], smart grid [7], intelligent transport system [8], etc., ICN scheme and architectural model can be utilized.

The research works on the on-path caching strategy of ICN can be divided in two broad categories. In the first approach, the content is cached in all routers of 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]. When a request arrives and a hit occurs at a specific level cache or 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 point 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 same content diminishes the probability of additional number of fetching of the same content, as long as it is in the proximity.

On the other hand, 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 in works [11], [12] due to its content redundancy and inefficiency in resource utilization. Since caches have limited memory, too many duplicates of the popular content results in less amount of space for other contents. Thus, reducing 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 neighbourhood 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 art of the ICN literature is as follows: here a content is copied at one or few routers (chosen randomly or strategically) of the path from content source to destination during content fetching. Works of ProbCache [11], MultiCache [13], Leave Copy Down (LCD) [10] etc., explains these type of category and serves as an example of it. However, the purpose of these studies is to reduce redundancy/replica in neighbourhood, this results in ending up the cost of increasing probability of fetching additional copy of the same content from repository. Otherwise stated, 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 support Name-Based Routing model where NDO information is stored and spreaded 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 where the Interest (NDO request) is forwarded by CCN router to other router or repository 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 and effective 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.

LEVERAGING THE NODE ARCHITECTURE, NAMING MODEL AND THE FORWARD ENGINE MODEL

In named data network (NDN), on-path caching approach is made. The work of NDN includes content centric networking (CCN) [19] proposal. We deploy the CCN node architecture and naming concept and the forwarding engine model. 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 interface of the Interest packets so that the data packet can be sent back through the same path towards the requester. The Interest is checked in 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.

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. However, 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 structure namely: the CS, the PIT and the FIB. 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.

Thus, when an entry arrives at the route/path towards the cache, the Interest will be redirected towards the (cache) node where there is a chance that the requested object may be found. This method of receiving data from the cache (which is not en-route) is defined as Exploring Phase in this paper. However, if no entry is found at the Cache-Route Table for the corresponding content the Interest will be forwarded to the face towards the permanent copy/repository. The method of directing information towards the permanent known copy of the content is defined as Exploiting Phase in this paper. The following section elaborates the forwarding engine mechanism of Exploring Phase and Exploiting Phase. The Exploring Phase is executed for a single time before the Exploiting Phase initiates.

PROPOSED CACHING AND FORWARDING STRATEGY

Forwarding 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 a situation. High level view of forwarding engine of our proposed scheme is shown in Fig.1.

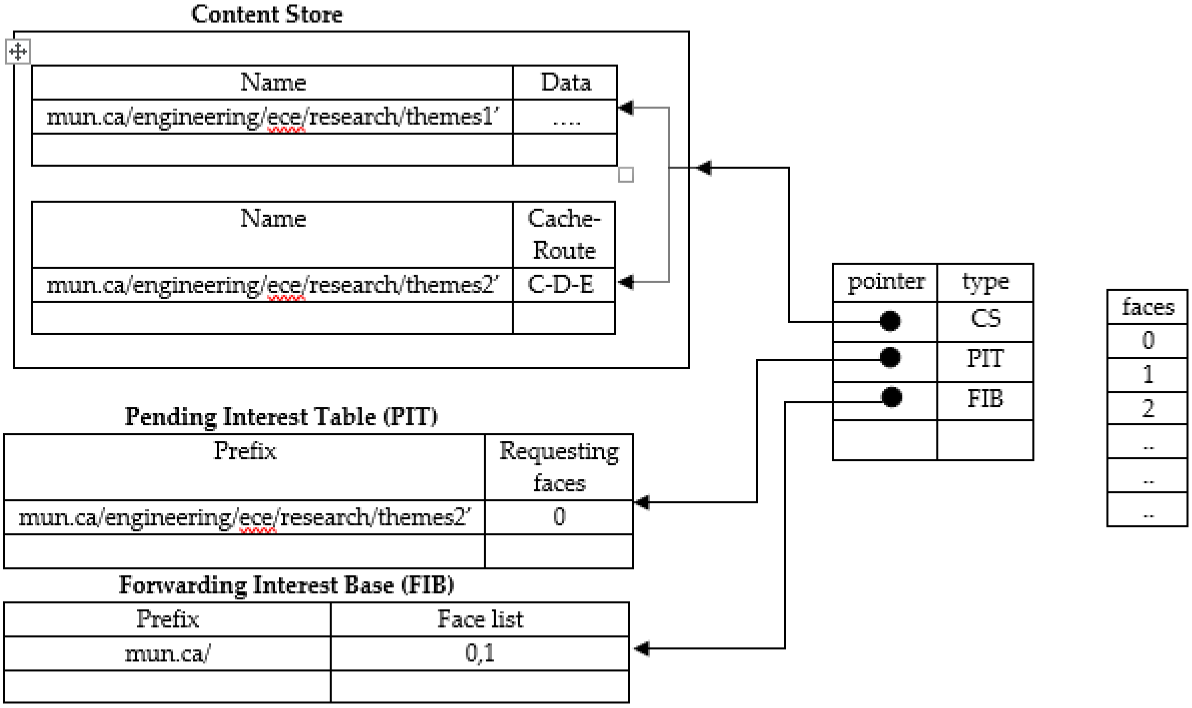

Fig.1 High level view of forwarding engine of Cache-Route strategy

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. Otherwise, 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 (Fig.1). 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.

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 forwarded 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 not have the property to have information of address of other routers except their 1-hop neighbours. 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 analysing 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.

Caching

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. 2-6 [20] and also pseudo code of the algorithm is given in Table-2.

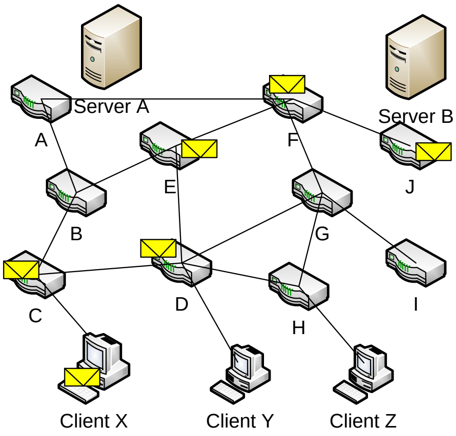

Fig.2 Local Connectivity of AS's Routers

In Fig.2, let, a repository adjacent to J claims that it can provide Interests matching the prefix'mun.ca/engineering/ece/research/'. The router J broadcasts the prefix 'mun.ca/engineering/ece/research/' in prefix announcements using CCN TLV [21]. 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 J 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.

Let the content '/mun.ca/engineering/ece/research/themes' requested by the client X 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 Fig.3 and Fig.6). 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 Fig.4-6). 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 Dare not larger than the repository distance from E 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.

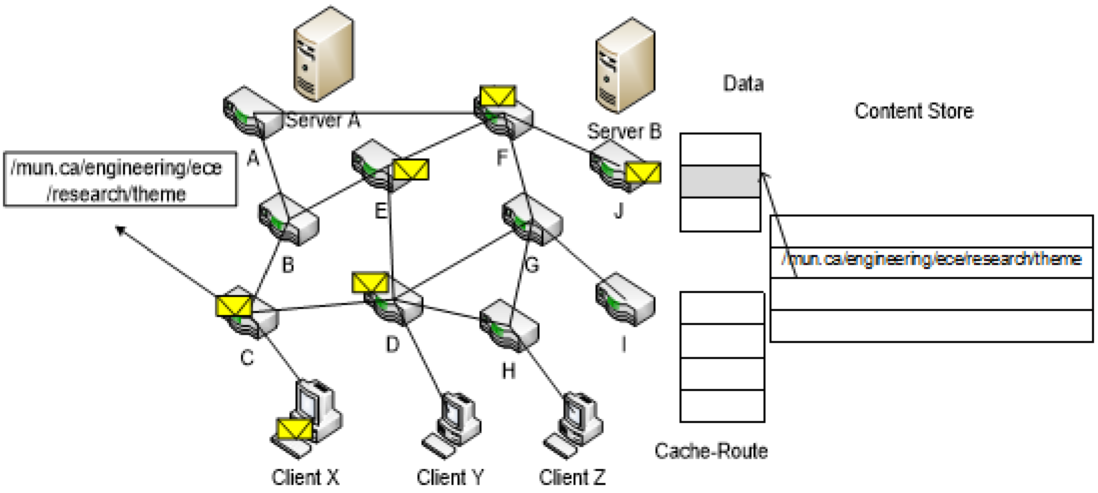

Fig.3 Content Store of Router C after retrieving the content ‘mun.ca/engineering/ece/research/themes’ from the repository next to J

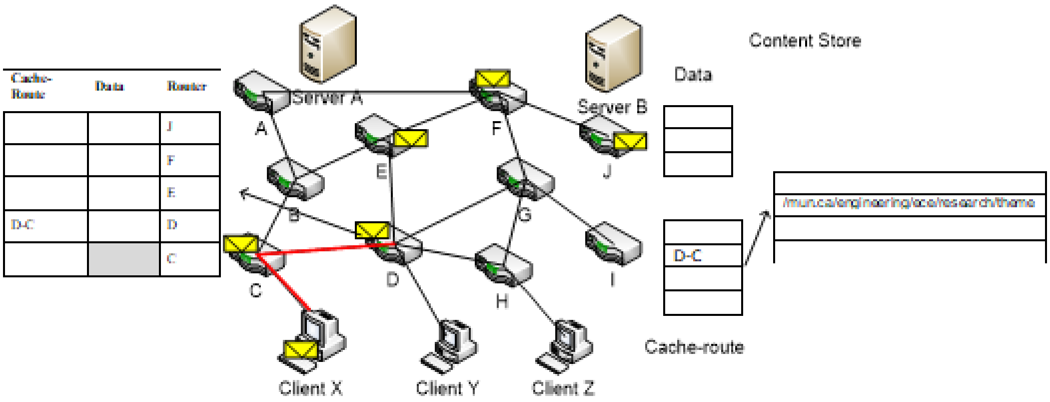

Fig.4 Content Store of Router D after retrieving the content ‘mun.ca/engineering/ece/research/themes’ from the repository next to J

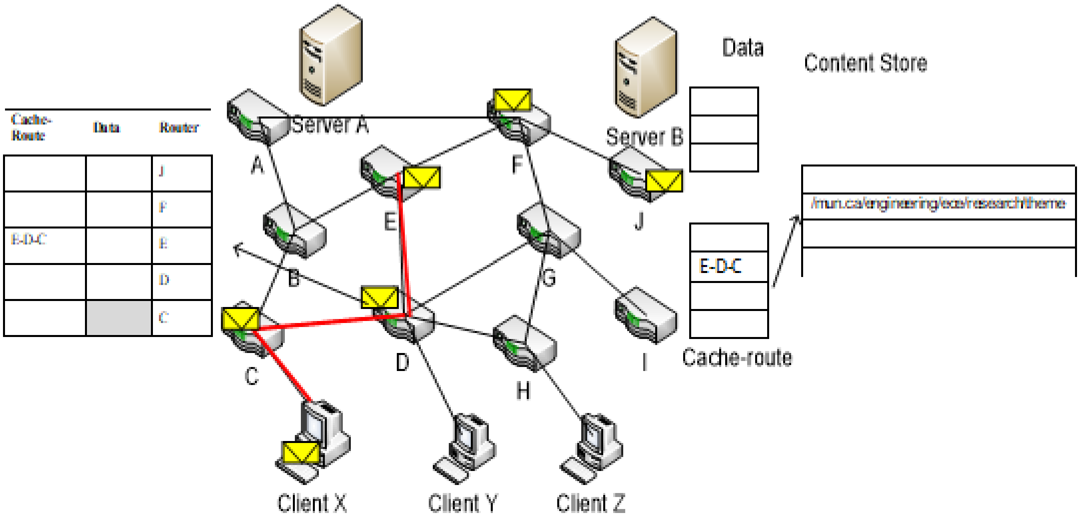

Fig.5 Content Store of Router E after retrieving the content ‘mun.ca/engineering/ece/research/themes’ from repository next to J

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

Fig.6 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, 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 (Fig. 3). 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.

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

Table-2 Pseudo Code Representing Cache Eviction Algorithm

Pseudo-code for forwarding Interest towards data Source(upstream)

```
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
Initialize (CR ←i)
Initialize (Explored ← 0)
WHILE (CR != j)
begin
  IF (data is found in Data table of CS)
      return CR
END IF
ELSEIF(Explored!= 1) and (data temporary
     location,T  is found in  CRT of CS)
       WHILE (CR != T)
        begin
          Explored ← 1
          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
```

Pseudo-code for forwarding Data from source to requester (downstream)

```
C: Source node (Cache/temporary copy) of the
   requested data
R: Source node (Repository) of the  requested data
i:Node  where Interest is generated
S ← CR value retrieved from upstream Algorithm
While(CR != i)
   begin
     CR  ← Next hop towards i
forward Data to CR
       IFhop_count (R) ≥hop_count (C)
        Update CRT of CS//Update cache
                       knowledge
    end
IF (S = = R) Store data at CS(Data table) of i
END IF
IF (S = = C) Update CRT of CS
END IF
```

RESULT AND DISCUSSION

NUMERICAL ANALYSIS

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 [22].

Let, L is the random variable corresponding to the fraction of request served by node received by the clients connected to node. 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 [23] [24], we can simplify the load imbalance, $C_v[L]$ as follows:

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

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

$$p(i) = \frac{1/_{i^\alpha}}{H_n^{(\alpha)}}, \tag{2}$$

$$H_n^{(\alpha)} = \sum_{j=1}^{F} 1/_{j^\alpha}, \tag{3}$$

Where, $p(i)$, expresses the content popularity for object i, which follows a Zipf distribution [25], with exponent α and $H_n^{(\alpha)}$ is the rank of the i-th most popular file in a catalog of size F files. $H_n^{(\alpha)}$ can be approximated as follows,

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

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

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

Let r be 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 most popular cached content for the client attached to it, i.e., the node cached the content of decreasing popularity of rank 1,2,……. r, then we get,

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

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

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

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 attached to the node which is capable of serving the request. Secondly, the request is forwarded from other nodes of the network. In the latter case, the request comes through Exploiting Phase (path getting from FIB) or Exploring Phase (path getting from Cache-Route table).

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

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

The cache hit ratio π_i can be approximated based on the work [26] where the value of r_{ij} can be found from the equation (8). Therefore,

$$\boldsymbol{\pi_i = 1 - e^{-p_i r_{ij}}}. \tag{9}$$

SIMULATION RESULTS

Simulation is done with ccnSim [27] which is a scalable chunk-level simulator of CCN under the OMNeT++ framework [20]. We consider the similar topology stated in [28]. 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 λ ≤ 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:

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.

Fig.7a shows a differentiation of our proposed Cache-Route scheme with the LCE and LCD schemes. 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 replicas compared to LCE.

Server Hit Ratio

Reduction of the server hit ratio is another important objective of deploying widespread cache structure of ICN. In Fig. 7b, 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.

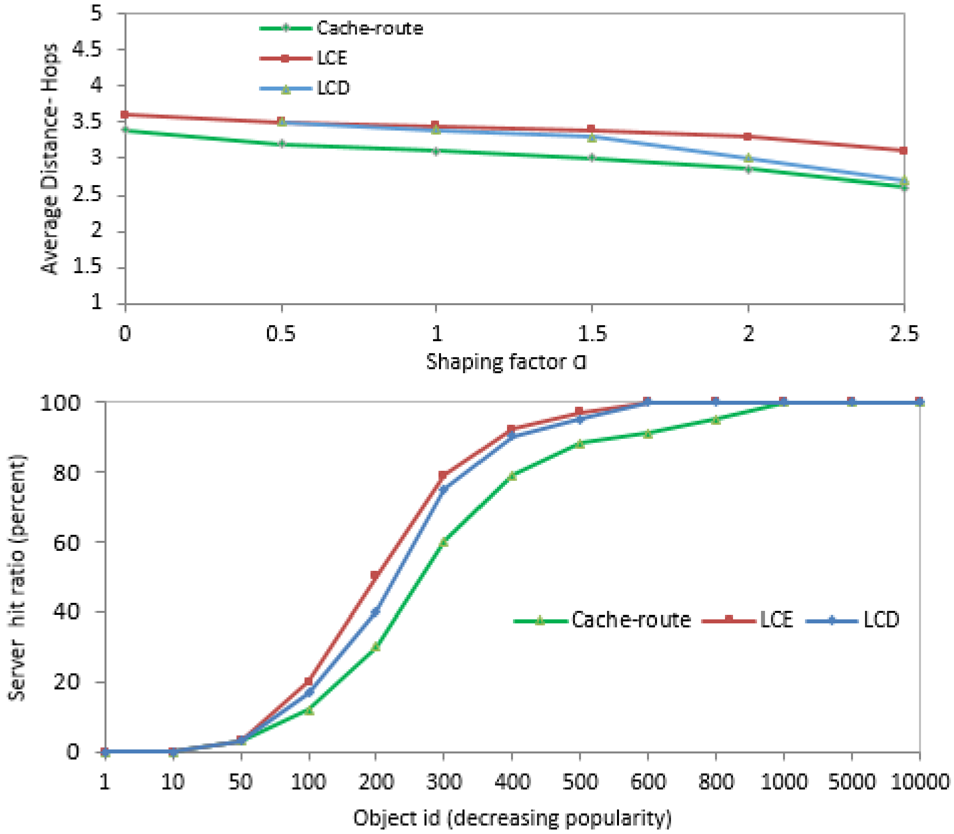

Fig.7 (a) Comparison of average hop-distance for Cache-Route strategy, LCE policy and LCD policy. (b) Comparison of Server/repository hit-ratio between Cache-Route strategy, LCE policy and LCD policy for the objects of different popularity

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 objects 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 are adopted in the process. We eradicate cache redundancy and 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.

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.

Acknowledgments
This work is supported by the EU-JAPAN co-funded Project Green ICN [29], (FP7 grant agreement N. 608518 and NICT Contract N. 167)

REFERENCE

[1] D Langutin, K Visala and S Tarkoma, Publish/Subscribe for Internet: PSIRP Perspective, *Towards the Future Internet Emerging Trends from European Research*, **2010**, 4, 75-84.
[2] V Jacobson, DK Smetters, JD Thorton, MF Plass, NH Briggs and RL Braynard, Networking Named Content, *Proceeding of 5th International Conference on Emerging Networking Experiments and Technologies, Co-NEXT* 2009, Rome, Italy, **2009**, 1-12.
[3] JE Lopez, M Arifuzzaman, L Zhu, Z Wen and S Takuro, Seamless Mobility in Data Aware Networking, *Proceeding of ITU Kaleidoscope: Trust in the Information Society*, Barcelona, Spain, **2015**, 131-137.
[4] PK Agyapong and M Sirbu, Economic Incentives in Information-Centric Networking: Implications for Protocol Design and Public Policy, *IEEE Communication Magazine*, **2012**, 50, (12), 18-26.
[5] M Arifuzzaman, K Yu and T Sato, Collaboration between Network Players of Information Centric Network: An Engineering-Economic Analysis, *Journal of Information and Communication Technology,***2015**, 2, 201–222.
[6] B Mathieu, P Truong, W You and JF Peltier, Information-Centric Networking: A Natural Design for Social Network Applications, IEEE Communication Magazine, **2012**, 50, (7), 44–51.
[7] K Yu, M Arifuzzaman, Z Wen, D Zhang and T Sato, A key Management Scheme for Secure Communications of Information Centric Advanced Metering Infrastructure in Smart Grid, *IEEE Transactions on Instrumentation and Measurement,***2015**, 64, (8), 2072–2085.
[8] Q Nguyen, M Arifuzzaman and T Sato, Proactive-Caching Based Information Centric Networking Architecture for Reliable Green Communication in Intelligent Transport System, *Proceedings of ITU Kaleidoscope: Trust in the Information Society*, Barcelona, Spain, **2015**, 139–145.
[9] X Tang and ST Chanson Coordinated En-Route Web Caching, *IEEE Transactions on Computers,***2002**, 51, (6), 595–607.
[10] N Laoutaris, H Che and I Stavrakakis, The LCD Interconnection of LRU Caches and its Analysis, *Performance Evaluation,***2006**, 63 (7), 609–634.
[11] I Psaras, WK Chai and G Pavlou, Probabilistic in-Network Caching for Information-Centric Networks, *Proceedings of 2nd ICN Workshop on Information-Centric Networking*, Helsinki, Finland, **2012**, 55–60.
[12] M Zhang, H Luo and H Zhang, A Survey of Caching Mechanisms in Information-Centric Networking, *IEEE Communications Surveys&Tutorials,***2015**, 17(3), 1473–1499.
[13] K Katsaros, G Xylomenos and GC Polyzos, MultiCache: An Overlay Architecture for Information-Centric Networking, *Journal of Computer Networks,***2011**, 55 (4), 936–947.
[14] R Chiocchetti, D Rossi, G Rossini, G Carofiglio and D Perino, Exploit the Known or Explore the Unknown? Hamlet-Like Doubts in ICN, *Proceedings of 2nd ICN Workshop on Information-Centric Networking*, Helsinki, Finland, **2012**, 7–12.
[15] *http://www.netinf.org/sail-netinf/* accessed on August 26, **2017**.
[16] *http://www.comet-project.org/* accessed on August 26, **2017**.
[17] *http://www.ict-convergence.eu/* accessed on August 26, **2017**.
[18] V Jacobson, M Mosko, D Smetters and JJ Garcia-Luna-Aceves, Content-Centric Networking, *Whitepaper Describing Future Assurable Global Networks*, Palo Alto Research Center, Inc.**2007**, 1-9.
[19] *http://tools.ietf.org/html/rfc4971* accessed on August 26, **2017**.

[20] M Arifuzzaman, Y Keping, QN Nguyen and S Takuro, Locating the Content in the Locality: ICN Caching and Routing Strategy Revisited, *Proceedings* of *European Conference on Networks* and Communications (EuCNC), Paris, France, **2015**, 423–428.

[21] EG Coffman and PJ Denning, *Operating Systems Theory*, Prentice-Hall Englewood Cliffs, NJ, **1973**, ISBN 0-13-637868-4.

[22] BM Maggs and RK Sitaraman, Algorithmic Nuggets in Content Delivery, *ACM SIGCOMM* Computer Communication *Review*,**2015**, 45(3), 52–66.

[23] L Saino, I Psaras and G Pavlou, Understanding Sharded Caching Systems, *Proceedings of IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, San Francisco, CA, USA, **2016**.

[24] L Breslau, P Cao, L Fan, G Phillips and S Shenker, Web Caching and Zipf-like Distributions: Evidence and Implications, *Proceedings of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, INFOCOM, New York, NY, USA, **1999**, 1, 126–134.

[25] *http://arxiv.org/pdf/0705.1970.pdf* accessed on August 26, **2017.**

[26] *http://perso.telecom-paristech.fr/~drossi/index.php?n=Software.ccnSim* accessed on August 26, **2017.**

[27] *https://omnetpp.org/* accessed on August 26, **2017**..

[28] *http://www.greenicn.org/* accessed on August 26, **2017**.

[29] *http://www.fp7-pursuit.eu/PursuitWeb*/, accessed on August 26, **2017.**