International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Connected Dominating Set Theory Assisted Caching Joint Routing Protocol for Information Centric Networks

Krishna Delvadia*[1]

Submitted: 07/01/2024 **Revised**: 13/02/2024 **Accepted**: 21/02/2024

Abstract: Information centric networks change the focus of existing network architecture from host driven to data driven model. It supports in-network caching and data driven forwarding which helps to reduce content retrieval delay, network congestion and network load. To forward interest packet so that desired data can be retrieved in least latency is key challenge for ICN. The routing strategy can operate efficiently if caching and request forwarding decisions are executed in a collaborative way. We introduce a collaborative approach between caching and forwarding in ICN to minimize latency. The caching mechanism exploits connected dominating set theory to avoid redundancy in caching and to reduce delay. The forwarding mechanism exploits content placement information as well as additional parameters like pending interest table (PIT) size of content router and total count of data packets originated by content router. This will help forwarding plane to send request to router having highest probability of having required content. The protocol uses the Dijkstra's shortest path routing algorithm and functions co-operatively with caching integrated forwarding strategy. The protocol performance investigation is carried out in ndnSIM-2.0 simulator using performance measures like content retrieval delay, cache store hit ratio and overhead. The performance analysis shows that proposed protocol outperforms the existing approaches for realistic network scenarios (GEANT and US-26) and exhibits enhancement up to 8-37% in mentioned performance metrics.

Keywords: *Information centric networks, Forwarding, Caching, Routing, Content centric networks*

1. Introduction

The networking domain has undergone a major transformation in past decades because of a tremendous growth of data. There will be significant requirement of greater throughput as well as minimal retrieval delay for future communication network [1]. Consumers will be concerned about quick retrieval of content without worrying about its source. Information centric network (ICN) is an emerging networking architecture that is designed to satisfy future consumers' needs. ICN supports in-network caching to minimize network congestion and data retrieval latency. Requester can retrieve desired content by sending request packet containing unique data identifier to network. The intermediate node having requested data can reply with corresponding data packet without only relying on content producer. The conception of ICN has been defined in various paradigms [2]. In addition to unique naming of data and caching of data in content routers, content identifier driven routing is also an important research challenge in ICN [3,4]. Routing and caching are performed based on content names instead of location dependent IP addresses. Following this notion, generate a content interest with identifier of desired content. The interest is then forwarded towards the suitable cache store based on forwarding protocol. ICN routing has mainly two important functions: interest forwarding towards suitable cache store and data forwarding back to the consumer node.

Each content router refers to forwarding information base (FIB) table to take interest forwarding decision. Majority of routing proposals emphasize on constructing routing information and populating the same in FIB. Though, some of the proposals focus on forwarding content interest to closest cache source without worrying about construction of FIB [5]. This approach functions as an overlay mechanism and helps to locate desired data in minimal time with lesser network overhead. Proposed protocol also functions as an overlay solution and does not focus on FIB construction. We have introduced a caching integrated routing solution for ICN with the aim of minimizing data discovery latency as well as overhead and maximizing cache hit ratio. The forwarding strategy exploits the content placement details to send request to a node having highest probability of containing requested data. The concept of connected dominating set (CDS) is used for creating virtual backbone network. The conception of CDS is used to identify the potential cache stores for content caching and this information will be used by forwarding module to redirect interest to node with greater cache hit probability. Forwarding module will

[1] *Department of Computer Engineering, Shree Swami Atmanand Saraswati Institute of Technology, Surat, Gujarat 395006*
** Corresponding Author Email: krishna.delvadia@ssasit.ac.in*

also exploits parameters such as pending interest table (PIT) size of node and total count of data packets generated by router to reduce data retrieval latency.

The aim of this research study is to introduce a collaborative caching integrated request forwarding mechanism. The research contributions for the proposed mechanism can be summarized as below:

- A connected dominating set driven caching strategy that create virtual backbone layer for given ICN topology comprising of controller content routers with greater likelihood of cache hit.
- A forwarding mechanism with aim of retrieving content in minimal time. It utilizes the content placement information derived through caching scheme and two other factors: pending interest table (PIT) size of content router and number of data packets originated by content router.
- In-depth performance investigation of protocol with comparative analysis using ns-3 driven ndnSim simulator [6] is performed.

Forwarding content interest to the cache store with greater likelihood of cache hit rate allows forwarding plane to fetch content with minimal latency and overhead. The integrated caching mechanism significantly contributes to identify such potential cache store. In contrast to proposed protocol, majority of existing proposals [7, 8, 9, 10, 11, 12, 13, 14] have developed a stand-alone and non-collaborative protocol either for caching or routing in ICN. Hence, ICN's in-network caching potential remains unutilized which result in greater data discovery latency [15]. The proposals were inclined more towards identifier-driven shortest path routing and have used distinct caching protocols to reduce data discovery delay. The proposed approach focuses on designing an effective co-operative caching integrated routing strategy which can minimize the data discovery delay and maximize the routing protocol's scalability. The strategy can be integrated to any state-of-the-art routing approaches. Its performance is comparable to existing routing mechanisms with different variants of caching protocols. The novel approach behind our protocol is that the notions such as PIT size of CR, total count of data packets generated by CR and CDS for network backbone formation haven't exploited jointly together till date for cooperative caching integrated interest routing in ICN as far as we are aware.

The remaining research paper is structured as follows: The state-of-the-art research study is discussed in section 2. Proposed approach followed with algorithms of caching integrated routing strategy is presented in section 3. The performance evaluation of protocol and in-detail comparative outcome analysis is discussed in section 4. Research work is concluded in section 5 along with discussion on future research directions.

2. The state-of-the-art research studies

The conception of connected dominating set theory is widely adapted as a prominent way to improve network performance for wireless scenarios. We have used CDS notion to differentiate potential CRs for caching in ICN and converted flat network scenario into hierarchical scenarios. It helps collaborative caching mechanism to function in more straightforward way. OSPFN [8] protocol can be considered as a first attempt in the field of routing in ICN. It extends existing link state routing (LSR) strategy to compute paths and distributes content identifiers. It does not support dynamic multipath forwarding and only provide an overlay solution for identifier driven data forwarding. NLSR [9] is another variant of state-of-the-art LSR algorithm that adapts hierarchical trust system, naming and specialized mechanism to support routing update dissemination and multipath routing. It does not provide support for inter-domain forwarding and its performance need to be investigated on realistic network topologies. The performance of above protocols has been tested on NDN testbed. The authors in [16] have implemented caching and routing strategies based on rendezvous points. It forwards interests for identical data chunks to same cache store. Due to this, there is less probability for explosion of data structures preserved by content routers. The above mechanisms have tried to reduce data discovery delay with no enhancement in cache store hit rate, network load and overhead.

The performance testing for these protocols yet needs to be done for real-time network topologies and on NDN testbed. A bloom filter assisted interest forwarding joint FIB construction strategy is presented by researchers in [13] with aim of reducing data discovery delay. The research work presented in [15] emphasizes on improving ICN routing performance by notifying forwarding engine about caching information and data placement within network. The authors in [17] have analysed ICN forwarding strategy as a bargain game with a motive of reducing data discovery latency and overhead. They have assumed buyer node as content provider/source and seller node as content requestor. Each intermediate router can be distinguished as competent or incompetent depending on its potential to fulfil requests. The competent one will be allocated some pre agreed trading price and incompetent one will be allocated a fraction of trading price. The above discussed mechanisms try to enhance the performance of existing name prefix-driven routing approach without exploiting benefits of NDN features. This approach contributes to greater data discovery latency and overhead. They focus on FIB construction. Though, our proposal aims to efficiently forward data requests to nearest cache source without worrying related to FIB building. This overlay mechanism contributes to reduced network overhead and data retrieval latency.

The machine learning driven strategy to identify attack named DDoS is introduced by authors in [18]. They have used different classification methods for traffic categorization. Realistic network scenarios have been used for dataset generation and functional testing in ndnsim. The researchers in [19] have implemented an automated method for detecting and reducing attack named cache pollution in ICN. Each content router acts as a reinforcement learning agent that is capable enough to differentiate between authorized and unauthorized data requests. The authors in [20] have designed a generative adversarial nets model integrated with deep learning to identify semantic aware pieces and also making it difficult to recognize. A support vector machine-based strategy is exploited in [21] to predict the cache hit probability for arrived data request. It eventually helps to maximize the cache hit frequency, throughput and minimize data discovery delay. A load balancing strategy driven by deep learning model for software defined networking-based content centric networks is designed in [22]. It utilizes an inevitable policy gradient mechanism to automatically be aware related to connection-weights by feature investigation of congestion flow. The above discussed research proposals have not utilized machine learning driven models specifically for routing in content centric networks instead they have exploited the same either in security or caching mechanisms. Hence, there is a need to utilize the strengths of machine learning models for routing in ICN. These state-of-the-art strategies are effective to retrieve desired data but still they generate significant amount of network overhead and greater content retrieval delay.

The researchers in [23] have proposed a co-operative caching strategy named PoolCache for ICN driven IoT networks. It aims to maximize the caching potential for some fixed clusters of routers. The strategy achieves reasonably good performance with minimal need of resources. The feasibility of this protocol for other future generation network scenarios is not analysed. An online caching and co-operative routing strategy is introduced in [24] to reduce the network load as well as latency and increase the utilization of cache store. The approach was more inclined towards enhancing exploitation of cache store with no focus on content popularity in near future. A co-operative content dissemination joint caching strategy for 5G network is introduced in [25]. To optimize energy in context of caching and data dissemination, a new energy utilization model is designed. Despite this, there is a significant research gap for an efficient method to control the communication interference and challenges related to heterogeneity of 5G scenarios. A co-operative caching method has been designed for content driven mobile sensor networks by researchers in [26]. It emphasizes on node centrality and node's distance from data source. The approach suffers from greater data discovery latency as they have not utilized caching details while taking request forwarding decision in network. The authors in [27] have improved the performance of network coding driven information centric network with the help of dynamic off-path cache investigation based multipath routing strategy. This research focuses on best exploitation of multipath routing capability for effective utilization of off-route stored content chunks. They have not focused on energy requirements of protocol hence a comparative trade-off among data delivery performance and energy efficiency is required.

2.1 Review analysis

The above reviewed existing strategies are inclined more towards interest forwarding with no exploitation of ICN's in-network caching potential. Being a key research area in ICN, an effective

integration of caching and interest routing has not been adequately resolved and comparatively less analyzed. The proposed caching strategy can minimize the caching duplication as the content will be stored only in central routers in between the route from hitting router towards consumer. The routing strategy focuses on retrieving data via shortest possible path using data chunk placement details. An effective route building up to the required data can noticeably reduce overhead as well as content retrieval latency. This derived important fact has given us the motivation to pursue research in the said domain of ICN. The proposed approach emphasizes more on designing a smarter routing mechanism by exploiting data chunk placement details to increase the efficiency of ICN forwarding protocol. Therefore, proposed collaborative strategy can be integrated to any existing strategy for ICN routing.

3. Proposed work

3.1 System model

The implementation of proposed protocol follows Named Data Networking (NDN) paradigm of ICN. Each Content Router (CR) holds three different repositories named forwarding information table, pending request repository and cache store. For retrieval of desired data, consumer produces a request packet with content identifier. The request is then routed in network to retrieve corresponding data packet. Any intermediate CR holding the desired data in cache store can respond to request by sending data packet in the reverse path from hitting cache to requestor. The research aim of proposed collaborative mechanism is to efficiently route the content interest in a manner which can fetch data with least network overhead and latency. We have assumed a network with N number of CRs; each with capability to cache/store M number of data chunks from D number of different data sources. The data chunk size is considered as B bytes. The consumer node can generate interests with frequency F per second. The request packets adhere the standard independent reference model. Our protocol exploits shortest path routing as a base mechanism to route interests within network and executes on top of it. Every CR uses the proposed collaborative caching strategy to cache data chunks. To replace old cached data chunks, cache replacement technique named LFU is used. The data packet follows an exact but reverse route of request packet transmission. The key objective of our protocol is to offer an extension of existing routing mechanism with a collaborative caching integrated forwarding mechanism in order to minimize data discovery latency. The protocol functions well with existing routing protocols as it requires least integration changes. This results in easier deployment and testing of proposed protocol on NDN Testbed.

3.2 Collaborative caching mechanism

Our protocol exploits the notion of connected dominating set (CDS) theory for creation of virtual backbone in given ICN topology. It divides the whole network in smaller sub networks. It has been majorly used in mobile and wireless topologies to form fixed virtual backbone infrastructure. The proposed collaborative strategy will operate on top of the virtual backbone created by CDS. This approach helps in minimizing the overhead and increasing the protocol's reliability. The underlying backbone infrastructure contributes in reduced search domain of routing and associated control overhead. This leads to rapid convergence of routing algorithm. The straight-away advantage of this backbone network is the fact that the size of sub network turns into smaller one. Hence, there exists various active research groups involved in developing approximation strategies for minimum CDS. A remarkable research contribution is made in recent years for designing quality backbone infrastructures. This has given us motivation to exploit the strengths of CDS in ICN as well for performance enhancement of forwarding and caching strategies. An illustrative network scenario for network decomposition after CDS operation is presented in Fig. 1. The CDS formation strategy transforms a given network in hierarchical bi-level infrastructure. The content routers available within CDS are named as controller routers forming the top level of hierarchy. All other nodes are named as ordinary ones forming the bottom level of hierarchy.

3.3 Collaborative caching integrated interest routing strategy

The proposed collaborative mechanism contains two major schemes: caching as well as interest forwarding. Both are considered as two individual and separate strategies in majority of ICN literature. As per conventional workflow, the caching technique will be selected based on policy, and interest forwarding is driven by repository called forwarding information table. Our algorithm treats forwarding as well as caching as two deeply associated strategies. The content placement choices will be made by caching scheme with consideration of present details available about routing. In contrast, the proposed protocol takes interest routing decision by utilizing the information about data chunk placement.

3.3.1 Caching strategy

The main purpose for new caching strategy is to eliminate excessive data replication within network. To attain this, controller nodes save the data having greater popularity value. Because of this, a network can cache a huge range of content. The proposed caching strategy can be articulated as follow:

- Proposed approach contradicts with existing on route caching strategy where content is stored in each router on its route from hitting content router towards consumer. Whereas in case of proposed caching method, only controller nodes along its route from hitting router to consumer store the desired data. The ordinary nodes along its way will not store the queried data.
- The controller node adopts the least frequently used strategy for replacing cached data chunks when the cache store size exceeds to its capacity.
- The content that is deleted from the controller router will be randomly sent towards any one of its ordinary neighbour nodes situated at one hop away within network segment. An index repository will be managed at controller node. When a controller router evicts any content, the related record will be inserted in its index repository. Index repository stores three key records (data chunk identifier, egress content router (CR), cache hit likelihood). The first record denotes the unique identifier for content chunk; second record denotes the name of ordinary node towards which the deleted chunk is sent; third record denotes the probability approximation for cache store hit of egress CR.
- The ordinary nodes exploit LFU as a cache replacement strategy for replacement of the data chunk and discard the removed content chunk immediately.

The popular data chunks are pushed towards controller routers and less popular chunks are pushed to ordinary nodes. To reduce caching replication, the protocol guarantees that the repeated data chunk is not available within controller CRs and ordinary CRs during the same instance of time. An index table is managed by controller CR so that subsequent request packets can be redirected towards corresponding egress content router as replica of that content chunk is already stored at one hop far router cache. For a requested data chunk, if there are more than one index records related to distinct egress content routers, then the controller router chooses the router having highest cache hit likelihood to forward request packet. For cache hit likelihood prediction for requested data at chosen router by controller router, we have exploited Che's approximation model [15]. We have shown an example of index record maintained at any controller router in Table 1.

Table 1. Index record structure

Data chunk identifier	**Egress content router**	**Cache hit likelihood**
Courses/B.E/Guidelines.txt	CR_6	0.88

This index record structure describes that controller router CR_5 has removed the data chunk Courses/B.E/Guidelines.txt and sent it to CR_6 having value of cache hit likelihood as 0.88 for CR_6. This cache hit likelihood prediction record of a specific content and egress CR entry is discarded when CS size of ordinary CR exceeds its capacity and it removes that data chunk depending on LFU to store fresh data chunks sent by sub network's controller router. This helps the protocol to minimize the computational overhead generated because of cache hit likelihood prediction calculation by controller router but only once that doesn't require periodic changes.

3.3.2 Interest forwarding

The traditional forwarding strategy exploits forwarding information repository to forward interest packet inside network. The introduced collaborative forwarding strategy uses this repository as well as exploits information about cached content. In addition, it refers additional details like pending interest table (PIT) size of node ($size_P$) and total counts of data packets generated through node (N_{DP}) to forward content interests towards CRs having highest probability of cache hits. This helps to significantly minimize the data discovery delay. An in-detail interest forwarding strategy is described in Fig. 1 (Refer Appendix 1). The key function of forwarding module is to first forward the unsatisfied requests towards controller routers of that particular network segment. The controller routers have highest probability of cache hits for required data as they store content with greater popularity. The controller router exploits an index table to save their ordinary CRs' information, such as their stored content (removed by controller CR). Hence, there is a greater likelihood of cache hit at controller nodes for given interest messages. If the interest is not answered by controller node, the ordinary CRs are queried based on index record maintained at controller CR. The forwarding strategy of protocol forwards the request to router having lesser value of $size_P$ and greater value of N_{DP}. This will happen when the controller node doesn't find any relevant index record of the required content chunk within its index table. The protocol focuses on fetching content via fastest/shortest feasible path to minimize data discovery latency and enhance efficiency of consumer level performance. The in-detail workflow for forwarding mechanism is described in Algorithm 1.

In case if any consumer needs data chunk, then it generates a request having the desired data's name in it. The request is then transmitted within network as mentioned in algorithm 1. Assume a case when any consumer 'C' produces a request 'R' for content chunk. With reference to proposed strategy, C forwards request to its direct neighbour CR. Upon receipt of request message, CR first verifies with its own cache if previously it has stored the required content chunk or not. If router has previously stored the identical data, then the relevant data message is directed in reverse to C; else current router is assessed for controller CR or ordinary one. In case of an ordinary CR, the request is sent towards particular network segments' controller CR. If it is a controller router, then the search is executed inside index table preserved at controller router. The index table stores the information about three key parameters as follows: Data chunk id, Egress CR, and cache store hit probability.

For the case when index table holds record for the required data, a request is sent towards associated egress CR; in other case, the algorithm exploits $size_P$ and N_{DP} values related to content routers inside network. Every router compute both these values and shares with its direct connected neighbours. Based on the received values, controller router prepares the local repository having three variables such as CR name, PIT size and Number of data packets originated. When a controller router needs to send a request upon unsuccessful search within its index table, it checks within its local repository and identify the router with highest N_{DP} and lowest $size_P$ values. The request is then sent towards that router. For a CS hit at such CR, a related data message is directed reverse back to the consumer. In case of cache miss at such CR, request is straight-away sent to the origin server through the fastest route with reference to FIB table. When the desired data gets found, it is forwarded back to the consumer. While data packet returns back from the hitting cache to the consumer, every in-between controller router caches the content chunk as directed by collaborative caching protocol. The significance of using the PIT size of node and Number of data packets originated by node is explained in the later subsection. The usage of these parameters in request forwarding process helps to significantly raise the cache hit probability at target content router for desired data chunk.

Algorithm 1: Interest forwarding mechanism

INPUT: CR named R_i receives a request I
E: Outgoing interface named I_O to target router chosen by protocol
$\boldsymbol{IS_{CNR}}$= Index repository at controller router R
$\boldsymbol{LR_R}$= Local repository table at router named R
$\boldsymbol{RMessage_{dc}}$=Request message to fetch content chunk *dc*
$\boldsymbol{DMessage_{dc}}$= Data message related to $RMessage_{dc}$
$\boldsymbol{SET_{CNR}}$= set of controller routers chosen by CDS mechanism
$\boldsymbol{Cache_R}$= Cache for content router named R

9 $\boldsymbol{R_{IT}}$= Index records at controller router named R
10 **START**
11 **If** $DMessage_{dc}$ in $Cache_R$ **then**
12 return $DMessage_{dc}$ towards consumer
13 **Else**
14 **if** R is present in SET_{CNR} **then**
15 Lookup for $RMessage_{dc}$ within IS_{CNR}
16 **if** $dc \in IS_{CNR}$ **then**
17 Send request to router having max{$L_{CS_{hit}}$} for dc
18 **Else**
19 Each router calculates its N_{DP} and $size_P$ values and shares the same with its immediate connected neighbour nodes
Controller router prepares LR_R for retrieved values of N_{DP} and $size_P$. LR_R={$R_i - N_{DP} - size_P$}; Here i=1 to n
20
21 Controller router searches for router having max{ N_{DP} }and min{ $size_P$}
22 Forward $RMessage_{dc}$to the router having max{ N_{DP} }and min{ $size_P$}
23 **if** dc is not present in $Cache_R$ **then**
24 ıd $RMessage_{dc}$ towards origin server
25 **end if**
26 **end if**
27 **lse**
28 Send $RMessage_{dc}$ towards controller router of that network segment
29 Go to step 14
30
31
32 **STOP**

a) Significance of $size_P$ (PIT size of CR) calculation:

The PIT table at any node holds the information related to unsatisfied interests travelled via that node. The minimal value of this parameter is desirable for receiving request packet. The reason for the same is that if node has already a large pending queue of unanswered interests, then there is less likelihood for either that node containing the desired data or be an intermediary to forward the needed content to observing node. So, from the point of view of request forwarding decision, a node with lowest PIT size is more trustworthy compare to remaining 1-hop neighbours of that specific node. In order to calculate this parameter, each node sends a query packet to its 1-hop neighbours individually to know its PIT size. After receiving query packet, each 1-hop node will answer this query with reply packet mentioning the PIT size of its own.

b) Significance of N_{DP} (Number of data packets originated by CR) calculation:

The number of data packets originated by CR is key parameter to analyse while taking request forwarding decision. Each node maintains a count for the total cache hit occurred at itself for requests. The node with higher value of this parameter is considered more reliable than others to forward given request packet. If any node has the higher count for this parameter, then there is high likelihood that it has the desired content in its CS. This also adds a direct contribution in minimizing the content retrieval latency. To compute this parameter, protocol introduces one additional field in each data packet that is returning back to requestor node. This field indicates about the content source for the needed data. Each received data packet at node contains the name of the content source, from where it has been originated. Each node maintains counter variables for each of its directly connected 1-hop neighbours. So whenever a node receives any data packet, it checks the data source name within packet. Consider a scenario when any node A has 3 directly connected 1-hop neighbours named B, C and D. now if A receives data packet via B, and content source of the same is also B then node A will increment the counter value for node B by 1. Likewise, it is applicable for C and D as well. Each node then sends a query packet to its 1-hop neighbours individually to know its N_{DP} count. After receiving query packet, each 1-hop node will answer this query with reply packet mentioning the cache hits count of its own. After receiving these values from neighbours, node will consider the next node having highest related counter value as more trustworthy than others to forward request packet.

4. Performance investigation

The experimental analysis of protocol using simulation investigates performance of shortest path routing coupled variants of LCD [28], CL4M [29], ProbCache [30], and LCE [31] with our co-operative protocol. The comparative performance analysis of proposed mechanism over other recent similar state-of-the-art strategies [13, 17] has also been carried out. A copy of requested content is kept within every CR on its way to content delivery path.

The LCD mechanism copies content in CR, located at one hop distance from requestor when cache hit happens. On other hand, CL4M protocol uses centrality measure of CR while taking decision related to caching. It results in minimized caching redundancy at different places. If the content is stored within router having greater centrality value, then the data accessibility among different consumers will also get increased. The protocol named ProbCache is based on the probability of retrieval request. It uses both the forecasted traffic frequency and on-path cache capacity to calculate the required likelihood of awaited interests based on popularity of content chunk. It supports caching of popular content chunks within interior routers. Therefore, CL4M protocol saves content only within particular routers on its way back to consumer. The ProbCahe mechanism predicts decision related to caching by selecting content based on probability of consequent access. The strategy presented in [13] aims to introduce a building strategy for routing table and FIB using an effective data advertisement mechanism in order to reduce overhead. They have also designed a request forwarding mechanism to minimize data discovery latency. The protocol proposal in [17] aims to reduce content retrieval latency and overhead by modelling forwarding problem as bargain game. As per the game rules, the reward points will be allocated to intermediate nodes based on their competency in serving interests and their support in construction of content transmission route. This analysis is useful to verify the efficiency of proposed protocol in forwarding requests to the nearest. the proposed collaborative caching integrated forwarding strategy can forward content interests towards closest suitable data source. It is the actual cause behind our investigation of the protocol performance compared to recent identical existing mechanisms.

Simulation parameters	**Value/Range**
Network scenario	GEANT topology
Simulation area	400m $\times$ 400m
Caching mechanism	Leave Copy Everywhere
Cache replacement mechanism	Least Recently Used
Simulation duration (T)	1200 seconds
Data popularity dissemination	Zipf distribution
Data Popularity dissemination exponent α	[0.2, 1.2] for User Generated Content
Total data chunks (N)	N=5000
Data chunk size	2000 Bytes
Cache store size	50-250
Data interest model	Independent Reference Model (IRM)
Custodian capacity (S)	500 MB
Interest arrival frequency (R)	20 requests/sec
Bandwidth	bps

Table 2: Simulation parameters

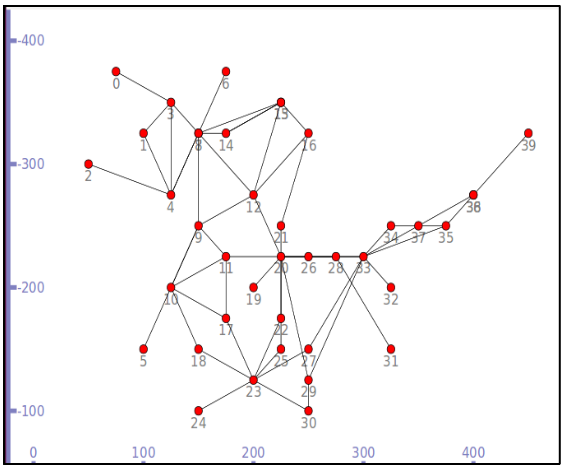

Fig. 2. The GEANT topology with 40 CRs deployed randomly over an area of 400 m × 400 m

4.1 Simulation environment

The experimental investigation of protocol is carried out in ndnSIM simulator [6]. The performance is tested for GEANT topology having 40 CRs (Fig. 2). Content routers are disseminated over a simulation area about 1000m × 1000m. The built-in ndnSIM environment exploits caching mechanism as LCE and cache replacement mechanism as LRU. Every simulation run has been given run time of 600 seconds. Applications are started after 15 seconds of gap after simulation run starts. Request packets adhere Zipf dissemination model with data popularity skewness co-officiant $\alpha \in [0.1, 1.3]$ as real-time values for User Generated Content [6]. The simulation variables and their corresponding values are represented within Table 2. For extensive investigation, the large set of values for α is considered. A uniquely differentiable group of 10000-18000 content chunks having size of each as 2500 Bytes are considered. During start time of simulation, the cache store is assumed empty. The capacity of content store is assumed in between 70-270, with a mean value about 80 data chunks. The custodian size S=600 MB, bandwidth=100 Mbps, and request arrival frequency R=25 interests/sec are assumed during simulation. We hypothesize that the protocol performance on real time network scenario in simulator show the similar performance on setup of NDN testbed as well.

1) Data discovery latency

The data discovery latency specifies the amount of time needed to discover the requested content chunk and send it back to consumer. It is always desirable to achieve lesser latency value for any information centric network. Here, we have analyzed the protocol performance in terms of latency for realistic GEANT network scenario. For each assessment of strategy, LRU is used for replacement of cached data chunks. The result analysis from Fig. 3 and Fig. 4 depict that proposed protocol remarkably outperforms existing approaches having fall in latency in range of 9-38% for GEANT network, increasing gradually with cache size. The performance of IF-ICN and BF-ICN nearly follows proposed protocol for $\alpha \in [0.6,0.8]$. For these α values, the fall in delay is 3-9%. The outcomes conclude the fact that the proposed protocol performs superior than state-of-the-art protocols. Though, in compare with LCD+SR, LCE+SR, CL4M+SR, ProbCache+SR, IF-ICN and BF-ICN, the latency reductions for our protocol are in the range of 10-37%, 20-48%, 12-36%, 15-25%, 6-12% and 4-10% respectively. Overall, in case of the GEANT scenario, the LCE+SR strategy has experienced the greatest latency to retrieve desired data as LCE caches every travelling data on every router. The IF-ICN and BF-ICN mechanisms outperform other existing approaches as it emphasizes on minimizing data retrieval latency by exploiting content placement details derived from their integrated caching mechanisms. The BF-ICN approach performs superior than the IF-ICN as in case of BF-ICN, the FIB construction is driven by protocol's own data advertisement mechanism.

The key reason for superior protocol behavior over existing approaches is that it utilizes data placement details while taking request forwarding decision. It aims to retrieve data through traversal of lowest count of nodes within network. The parameters like router's PIT size and number of data packets originated by CR are also exploited, in order to send request to node having higher likelihood of containing desired data. This leads to significant reduction in latency. The record of removed chunks from controller router is kept within index table, this helps to forward corresponding requests to particular egress router (having asked data) without sending request till custodian/origin server with the help of shortest path routing algorithm. Hence, collaborative forwarding mechanism uses alternate paths exclusive of shortest routes and attain cache hits at closest cache stores. In contrast with this, major existing protocols restrict themselves to shortest path routing approach.

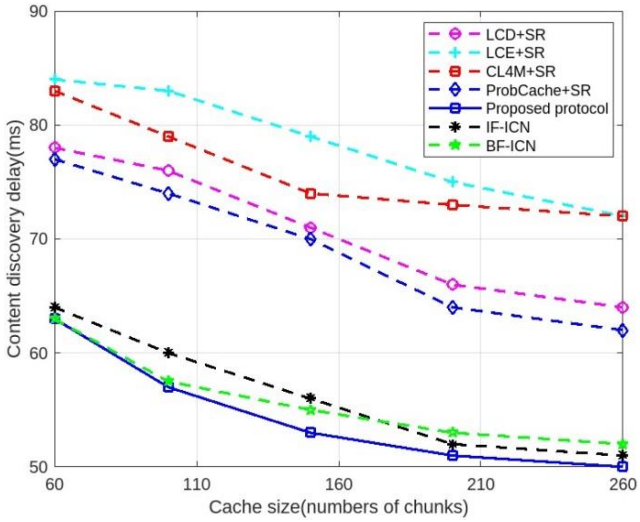

Fig 3: Cache size vs. content discovery delay for α=0.6

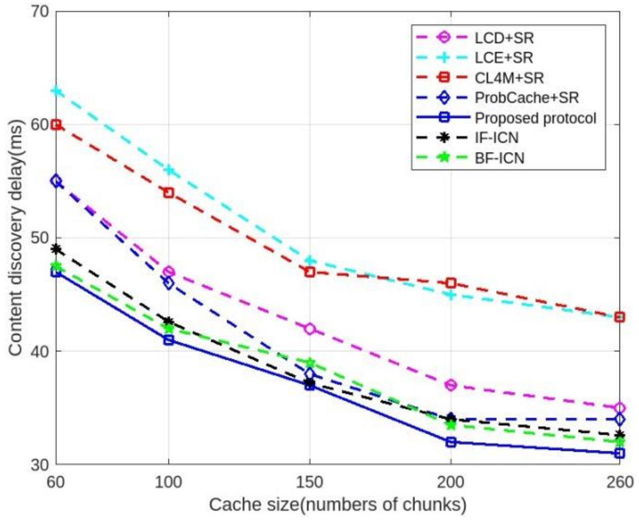

Fig 4: Cache size vs. content discovery delay for α=1.1

2) Network overhead

It is calculated as an overhead to network because of cache miss events in target router's cache selected by our protocol. The simulation outcomes in context of network load percentage changes over data request arrival frequency is represented in Fig. 5. The overhead value can be denoted with ratio of total requests resulted in cache misses at target CR over total count of requests originated in last averaging interval. The same notion can be expressed using following equation 1:

$$Overhead(\%) = \frac{\text{Number of requests result in CS miss at router}}{\text{Total requests originated in the last averaging interval}} \quad (1)$$

As this ratio value increases, the number of cache misses increases at target CRs. Due to this, an unanswered request will travel up to custodian to retrieve desired data. Hence, distance travelled by requests increases which generates the additional quantity of overhead for network. To calculate overhead, we store the distance traversed by each interest for that the cache miss happened at target CR. The fraction of changes for network load because of the need for content replacement is shown in graph. The request arrival frequency is linearly proportional with network overhead as represented in graph.

This ratio is used to denote the additional load incurred because of cache miss at suggested CR given by protocol. The rise in the ratio denotes the fact that number of requests that turn in cache miss at target CR, are increasing. As a consequence of this, the unsatisfied interests need to travel till origin server to fetch required content. So, the overall distance traversed by each interest in network increases as the interest has already travelled to target CR but desired content was not present. If a greater number of such CS misses events increases then the network overhead value also increases significantly. For network overhead calculation, protocol saves the distance travelled by each request packet for which the content access attempt to said router was unsuccessful. The fraction of updates in network load due to the requirement of the data replacement is depicted in graph.

The interest arrival rate is directly proportional to increased overhead as plotted in graph.

The performance of proposed protocol is compared with LCE+SR, CL4M+SR, LCD+SR ProbCache+SR, IF-ICN and BF-ICN mechanisms in context of overhead. Simulation results depict the mixed outcomes for all specified protocols. The proposed protocol outperforms existing protocols for different values of popularity skewness and cache size. Up to request arrival rate R = 25 requests/sec, our approach perform superior than LCE+SR, CL4M+SR, LCD+SR ProbCache+SR, IF-ICN and BF-ICN protocols by 6-13%, 7-17%, 8-16%, 9-21%, 5-11% and 3-9% respectively. The LCD+SR perform worst while LCE+SR and ProbCache+SR depict a comparative moderate performance till R reaches to 25. For this time, the network overhead of IF-ICN and BF-ICN over request arrival frequency R nearly adhere the behaviour of our algorithm. The fact for the better performance of BF-ICN over LCE, LCD, CL4M, and ProbCache protocols is that it sends request towards closest data source having highest probability of containing desired data. When the value of R becomes greater than 32, the outcome values of these algorithms also vary. For R > 36 requests/sec, LCE+SR protocol performs better over CL4M+SR, LCD+SR, ProbCache+SR. The cause for this is that LCE leads to dense caching of content inside network. On the other hand, in case of ProbCache, due to probability prediction misses, the cache miss happens frequently and generates a greater quantity of overhead. The BF-ICN protocol nearly follows the performance of proposed protocol for different R values. The rationale is that it utilizes the closest off-path stored encoded chunks and maintains the reachability details of off-path stored content at the generation phase of detailing. The moment R values exceeds to 36 requests/sec, our strategy performs superior than LCD+SR, ProbCache+SR, LCE+SR, CL4M+SR, IF-ICN and BF-ICN protocols by 10-17%, 9-21%, 9-14%, 8-20%, 11-22% and 6-11% respectively. The BF-ICN protocol generates smaller quantity of overhead in comparison with proposed protocol till R = 46 requests/sec. The strategy generates a noticeable overhead hike up to 11-19% after this time due to the use of add-on data advertisement mechanism for FIB building.

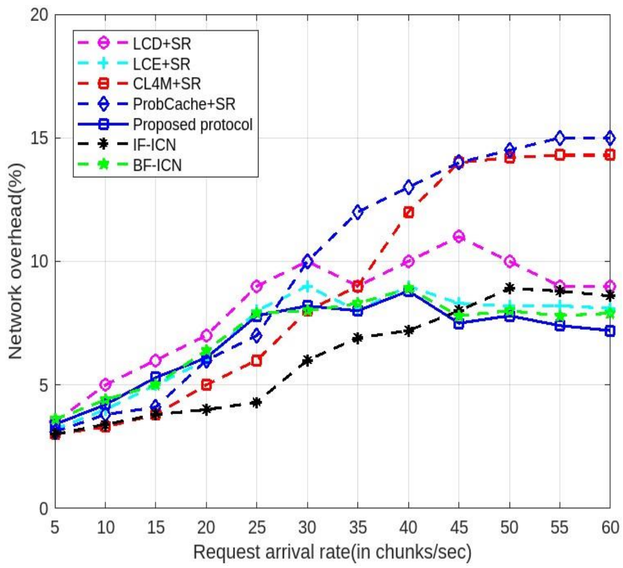

Fig 5. Request arrival rate vs. Network overhead

3) Cache hit rate:

This subsection analyses the algorithm performance over cache hit rate. During simulation, if request is satisfied from intermediate CRs, it is labelled with cache hit and if it is responded by custodian then it is termed as cache miss. Fig. 6 depicts the cache hit rate behaviour of our protocol and other existing proposals. We have calculated an average cache hit rate after taking into account different CS size values between 60 to 260 data chunks. We deduce the result that our protocol performs superior than existing approaches in terms of cache hit rate. The key cause for minimized latency of proposed protocol is that ICN tries to satisfy requests through intermediate caches instead of custodian. Therefore, to attain maximal cache hit rate for network is desirable performance goal in ICN. If the underlying request forwarding strategy effectively forwards a request packet, then it doesn't need to go till custodian, provided that at least one replica of content chunk is present within intermediate cache.

We have shown the result of the cache hit rate for a realistic GEANT network scenario over request arrival rate in Fig. 6. With rise in request arrival rate, the cache hit rate also gets increased for each discussed protocol. The Probcache+SR protocol performs well up to 9-16% and 7-14% in comparison with CL4M+SR and LCE+SR. The IF-ICN strategy performs identical with ProbCache+SR till request arrival rate R = 30 requests/sec. When value of R becomes greater than 30, the performance of IF-ICN and BF-ICN nearly follows the proposed protocol and continues with better performance, a significant good cache hit rate in comparison with other protocols. The fact behind a fair-enough performance of IF-ICN is that it utilizes monte carlo tree search algorithm for optimal route construction from requestor to data source itself. The BF-ICN protocol outperforms IF-ICN with 6-14% for R > 35 requests/sec as it aims to satisfy requests by interior routers first with the help of its information gradient driven efficient forwarding strategy. This result in noticeable hike in the cs hit rate of this protocol. The proposed protocol performs superior than CL4M+SR, ProbCache+SR, LCD+SR, LCE+SR, IF-ICN and BF-ICN by 9-18%, 8-15%, 10-21%, 7-19%, 6-13% and 8-11% respectively for cache capacityϵ [60,260] and different values of αϵ [0.3,1.6].

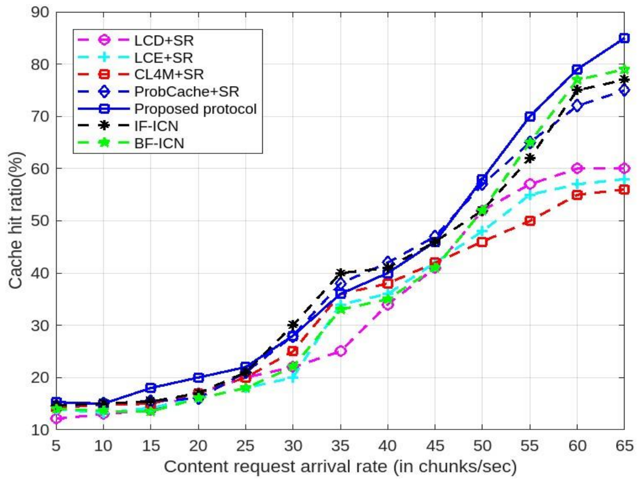

Fig 6. Content request arrival rate vs. Cache hit ratio

5. Conclusion

Proposed research work aims to introduce a co-operative caching integrated routing solution in ICN that can minimize the data discovery latency, overhead and maximize the cache hit rate. In ICN, if forwarding plane has information about caching details, then the requests can be forwarded to caches having higher likelihood of containing requested chunks. This results in significant enhancement in above specified performance measures as well. With inline to this objective, we have designed a caching mechanism using connected dominating set theory conception for ICN, on which the proposed new forwarding mechanism can run with assurance of significant reduction in latency and overhead values. We have tested protocol performance for GEANT and US-26 realistic network scenarios in ndnSIM simulator in comparison with existing caching protocols integrated with shortest path routing as well as with recent collaborative proposals in literature. The proposed protocol outperforms existing approaches in a range of 8-37% for performance measures like content retrieval delay, CS hit rate, network overhead.

CRediT authorship contribution statement

The corresponding author has worked in the formulation of problem, algorithm investigation, design as well as implementation work.

Declaration of competing interest

The author declare that she has no known competing financial interests to influence the work reported here. The author did not receive any funding support from any organization for the work. The data produced during simulation runs of proposed protocol are with author of the manuscript and can be available as requested.

Data Availability Statement

The data underlying this research will be shared on reasonable request to the corresponding author.

References

[1] M. Aggarwal, K. Nilay, and K. Yadav. Survey of named data networks: future of internet. *International journal of Information Technology (Singapore)* 2017; 9:197–207.

[2] H. Jin, D. Xu, C. Zhao, and D. Liang. Information-centric mobile caching network frameworks and caching optimization: a survey. *Eurasip Journal on Wireless Communications and Networking* 2017; 33:1-32.

[3] G. Xylomenos *et al.* A Survey of information-centric networking research. *IEEE Comm Surveys and Tuts* 2014; 16:1024-1049.

[4] F. Khandaker, S. Oteafy, H. S. Hassanein, and H. Farahat. A functional taxonomy of caching schemes: Towards guided designs in information-centric networks. *Computer Networks* 2019; 165:106937.

[5] T. Bektaş, J.-F. Cordeau, E. Erkut, and G. Laporte, ‘‘Exact algorithms for the joint object placement and

request routing problem in content distribution networks,'' Comput. Oper. Res., vol. 35, no. 12, pp. 3860–3884, Dec. 2008.

[6] R. Chiocchetti, D. Rossi, and G. Rossini, ''CcnSim: An highly scalable CCN simulator,'' in Proc. IEEE Int. Conf. Commun. (ICC), Budapest, Hungary, Jun. 2013, pp. 2309–2314.

[7] N. Dutta *et al*. Deep learning inspired routing in ICN using Monte Carlo Tree Search algorithm. *Journal of Parallel and Distributed Computing* 2021; 150:104-111.

[8] Lan Wang *et al.* OSPFN: An OSPF Based Routing Protocol for Named Data Networking. *NDN Technical Report NDN-0003, July 2012*.

[9] A. K. M. Mahmudul Hoque *et al.* NLSR: Named-data link state routing protocol. *In Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*. China, pp. 15-20, 2013.

[10] J. V. Torres *et al.* An autonomous and efficient controller-based routing scheme for networking Named-Data mobility. *Computer Communications* 2017; 103:94-103.

[11] C. Ghasemi *et al*. MUCA: New Routing for Named Data Networking. *In IEEE IFIP Networking Conference (IFIPNetworking) and Workshops*, Switzerland, pp. 289–297, 2018.

[12] X. Chen *et al*. Improving NDN forwarding engine performance by rendezvous-based caching and forwarding. *Computer Networks* 2018; 145:232-242.

[13] N. Dutta. An approach for FIB construction and Interest packet forwarding in information centric network. *Future Generation Computer Systems* 2022; 130:269-278.

[14] K. Delvadia, N. Dutta, and G. Ghinea. An efficient routing strategy for information centric networks. *In Proc. IEEE ANTS*, India, pp. 1-6, 2019.

[15] K. Delvadia, N. Dutta, and R. Jadeja. CCJRF-ICN: A Novel Mechanism for Coadjuvant Caching Joint Request Forwarding in Information Centric Networks. *IEEE Access* 2021; 9:84134 - 84155.

[16] X. Chen et al. Improving NDN forwarding engine performance by rendezvous-based caching and forwarding. Computer Networks 2018; 145:232-242.

[17] N. Dutta. A bargain game theory assisted interest packet forwarding strategy for information centric network. *Journal of Network and Computer Applications* 2023; 209: 103546.

[18] J. Bhayo *et al.* Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks. *Engineering Applications of Artificial Intelligence* 2023; 123: 106432.

[19] A. Hidouri *et al.* Q-ICAN: A Q-learning based cache pollution attack mitigation approach for named data networking. *Computer Networks* 2023; 235: 109998.

[20] R. Hamid *et al*. Deep reinforcement learning based Evasion Generative Adversarial Network for botnet detection. *Future Generation Computer Systems* 2024; 150: 294-302.

[21] N. Dutta et al. SVM-based Analysis for Predicting Success Rate of Interest Packets in Information Centric Networks. Applied Artificial Intelligence 2021; 36: 2020488.

[22] N.Rikhtegar, O.Bushehrian, and M.Keshtgari. DeepRLB: a deep reinforcement learning-based load balancing in data center networks. International Journal of Communication Systems 2021; 34:1-23.

[23] B. Alahmri, S. Al-Ahmadi and A. Belghith, "Efficient pooling and collaborative cache management for NDN/IoT networks", IEEE Access 2021; 9:43228-43240.

[24] K. Thar, N. H. Tran, S. Ullah, T. Z. Oo and C. S. Hong, "Online caching and cooperative forwarding in information centric networking", IEEE Access 2018; 6:59679-59694.

[25] Y. Sai, D.-Z. Fan and M.-Y. Fan, "Cooperative and efficient content caching and distribution mechanism in 5G network", Comput. Commun. 2020; 161:183-190.

[26] G. Jaber and R. Kacimi, "A collaborative caching strategy for content-centric enabled wireless sensor networks", Comput. Commun. 2020; 159:60-70.

[27] X. Hu, S. Zheng, G. Zhang, L. Zhao, G. Cheng, J. Gong, et al., "An on-demand off-path cache exploration based multipath forwarding strategy", Comput. Netw. 2020; 166:1-12.

[28] N. Laoutaris, H. Che, and I. Stavrakakis. The LCD interconnection of LRU caches and its analysis. *Perform. Eval*. 2006; 63:609-634.

[29] W. K. Chai *et al.* Cache 'less for more' in information-centric networks (extended version). *Comput. Commun.* 2013; 36:758-770.

[30] A. Psaras,W. K. Chai, and G. Pavlou. Probabilistic in-network caching for information-centric networks. In *Proc. ACM SIGCOMM ICN Workshop Inf.-Centric Netw.*, Finland, pp. 55-60, 2012.

[31] Y. He *et al.* A caching strategy in content centric networks based on node's importance. *Inf. Technol. J.* 2014; 13:588-592.

Appendix 1

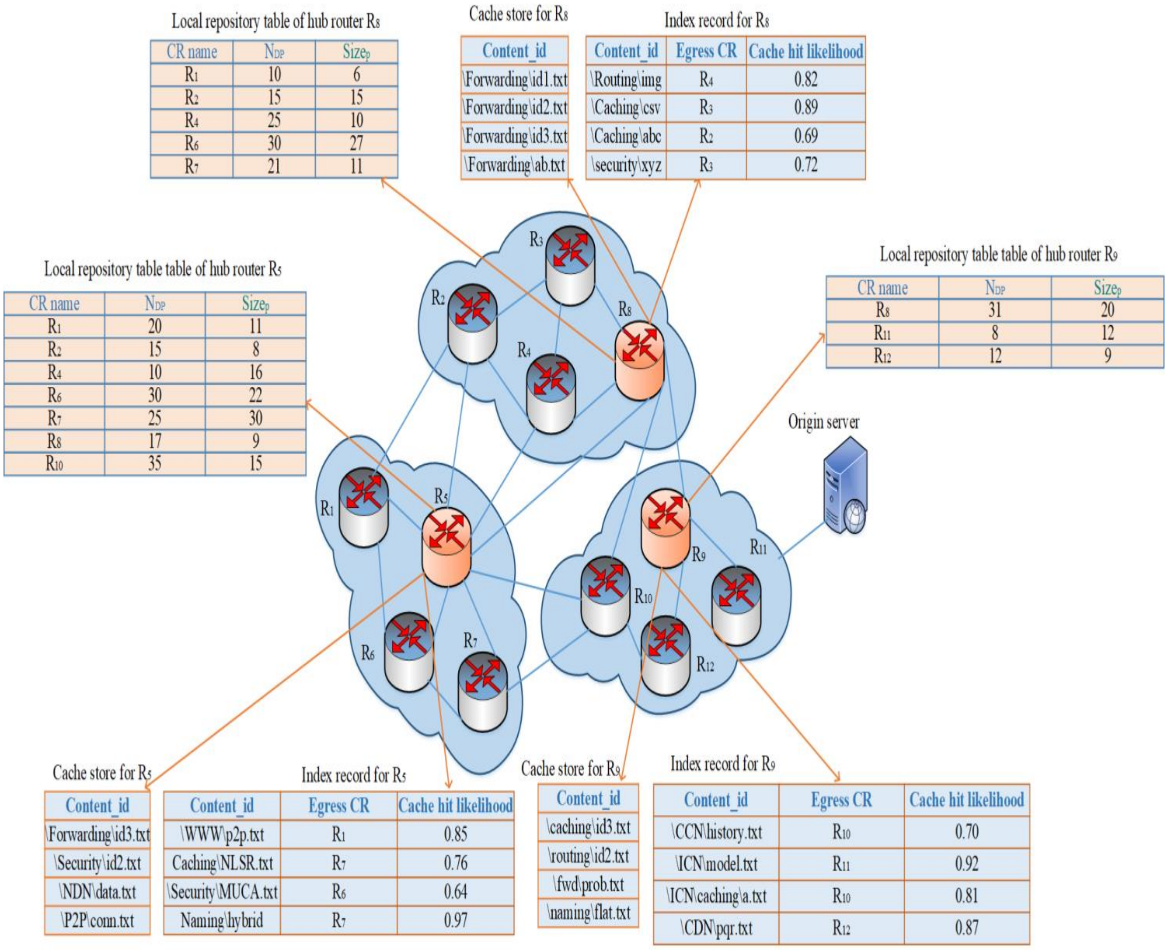

Fig 1: Network topology illustration for proposed collaborative protocol