This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/ANTS47819.2019.9118123 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)

An Efficient Routing Strategy for Information Centric Networks

Krishna Delvadia
Computer Engineering Department
Sarvajanik College of Engg. and Tech.
Gujarat, India
krishnadalasaniya10@gmail.com

Nitul Dutta
Computer Engineering Department
Marwadi University
Gujarat, India
nituldutta@ieee.org

Gheorghita Ghinea
Computer Science Department
Brunal University
London, UK
george.ghinea@brunel.ac.uk

Abstract-**Information Centric Network (ICN) focuses on shifting the current architecture of internet from host oriented to data oriented. It emphasizes on the location of contents rather than producer of the same. ICN accomplishes it's objectives by supporting caching of content at internal nodes in the network. It also focuses on proper routing of content requests towards a suitable data source for efficient and timely delivery. Hence, routing and caching are two prominent areas of research in ICN. In this paper, a dynamic routing mechanism is proposed that utilizes the concept of likelihood time and betweenness centrality. In order to improve content retrieval latency, the content requests are routed to the nearest content locations, determined by the likelihood time of requested content and betweenness centrality of a node in the network. The routing strategy of the proposed scheme extends native Dijkstra's algorithm and works with the available caching mechanism. The performance of the proposed strategy is evaluated through exhaustive simulations in ndnSIM-2.0, which is a ns-3 driven NDN simulator. The observed simulation results depict that in realistically simulated network topology, the new protocol shows 5-8% of performance improvement over existing ICN routing strategy in terms of cache hit ratio, latency and packet overhead.**

Keywords- ***Information centric networks, Routing, Betweenness centrality, Likelihood time, Latency 3011209715***

I. Introduction

The communication technology has undergone a considerable transformation in last couple of decades. This evolution is taking place in diversified directions including CCN [1], SDN [15], CRN [16, 17, 18] etc. The most promising changes have noticed in the Internet architecture. The current paradigm of the internet is built upon host-oriented communication model considering producer of data as the prime concern. In contrast, the analysis of current internet usage patterns depicts that it has evolved over the years and emphasized more on retrieving contents without concerning about its location [1]. This transition in usage patterns of the internet with persistent demands of users, fascinate researchers to innovate a radical change in existing internet architecture from host centric to content centric. As a consequence, the Information Centric Networks (ICN) have come into existence. This approach ascension certain challenges in ICN like naming of contents, routing [1], caching [2], security [3], mobility, trust, and support for scalable content distribution. Caching data at internal nodes the network empowers content delivery from those intermediate nodes instead of acquiring data form the distant original server. Such enrooting of content from intermediate cache significantly meliorates end-to-end delay and traffic load on the network [3].

One of the critical remonstrance in ICN is to design an efficient tactic for routing and caching that actually increase throughput and decrease intermission (latency or delay). There are many works reported in this promising area of ICN [4, 5, 6]. However, routing in ICN is not matured enough till date to afford a flawless solution for content routing. Considering the existing research gap, and demand of time, in this research, an efficient routing mechanism is proposed by exploiting the benefits of likelihood time and betweenness centrality (BC) of a node to minimize content retrieval delay. The proposed work also has a promising intermingling to improve cache hit ratio while sinking overhead. The noteworthy contributions of our research work may be enlisted as follows.

- Routing is centered around likelihood time and BC value of a node. It uses TTL value of a content to preserve the state of contents and its age.
- Intermediate routers maintain BC table, consisting of BC values of each of its 1-hop neighbors. It helps in routing with surrounding information and provides precise routing.
- An in-depth simulation in ndnSIM to prove the correctness and effectiveness of the proposed strategy over existing one.

An analysis of few distinct parameters of the proposed algorithm is carried out and compared with existing solutions presented in [8]. The simulation depicts an improvement of 5-8% in content retrieval delay, cache hit ratio and overhead compared to existing solutions. Our work differs from work presented in [8] in several ways. They have referred Che's approximation to find the probabilistic time, but we use extended Che's approximation with potential benefits over reasonable estimate as mentioned in [12]. The concept of BC of a node is extended to retrieve data in a fastest possible time.

The rest of the paper is structured as follows. The literature review of some recent work in the field of ICN routing is found in Section 2. The problem statement and proposed algorithm are discussed in Section 3. Simulation with result analysis is presented in Section 4. The paper is concluded in section 5.

II. Related Work

The routing in ICN deals with forwarding of interest and data packets defined by NDN architecture for ICN from one node to another in a network. The routing protocol in NDN [4] prepares the forwarding information base (FIB) table by inserting the path to the named content. The FIB act as a routing table in TCP/IP network to forward interest packet towards the data source. In this section, we briefly explored the existing routing related research with a highlight of differentiation with our approach.

Among different routing mechanisms available so far, OSPFN [4] is a basic and initial attempt towards routing in ICN. Authors have extended link state routing protocol to disseminate name prefixes and compute paths. It is

Copyright © 2019 Institute of Electrical and Electronics Engineers (IEEE). Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works by sending a request to pubs-permissions@ieee.org. For more information, see https://www.ieee.org/publications/rights/rights-policies.html

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/ANTS47819.2019.9118123 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)

currently being deployed in NDN testbed [4]. It provides routing based on names and has support for configured multipath routing. NLSR (Link State Routing for Named-data) [5] is another variation of intra domain link state routing procedure for NDN. It differs from IP based link state routing in the following ways. First, hierarchical naming, second, hierarchical security trust model, routing update propagation using ChronoSync protocol and fourth, multipath routing. It has also been deployed on the NDN testbed [4].

The hyperbolic routing (HR) described in [6] uses a greedy method to find content. The Nodes are modeled via coordinates in the hyperbolic space. The algorithm gives the shortest path in real-time with optimized use of computational resources. It also shows a stable topological behavior in case of frequent link failure and preserves load balancing. The work of [7] presents an addressing scheme with optimally scalable routing. The protocol needs the sizes of FIB to be same as number of adjacencies of a node. The protocol restricts the network to be internet-like topology only. This protocol can be mostly deployed in overlay networks. If the topology grows, it must follow the pre-defined design specifications.

The research work [8] aims to reduce content retrieval delay with the concept of characteristic time which is defined as the expected time duration in near future that content is expected to be in the cache from its last access. This time is used to send a request to the cache having a high probability of holding the content. The stated research has not emphasized on optimization of scope. Nor it uses granularity for local search in lookup tables of neighbors while corresponding content is not found in one's own lookup table.

The authors have proposed controller driven routing method [9] that will run on top of the NDN. It augments the actions of the router and reduces control and communication overhead between routers and controller. The protocol is not tested with respect to realistic internet topologies or with related routing approaches. It is not even explored for the cooperation of routing or caching controller. A variation of controller based routing scheme is proposed in [10] for NDN. It addresses the issue of FIB explosion by decreasing control message overhead. It is basically a CRoS runs on top of NDN. It adapts its features like controlling traffic, detect network problem and diversity of path. The authors have investigated tradeoffs among patterns of route expiration and caching.

MUCA [11] is a name based intra domain routing protocol with three major contributions. It decreases content retrieval latency and increases the resiliency of network while decreasing routing overhead. The protocol combines working of link state and distance vector routing in order to support multipath routing and to increase cache hit ratio. It utilizes in-built caching support in NDN to find new routes. It is a new method for synchronizing LSDB that makes MUCA better to NLSR in terms of performance.

Distance based content routing (DCR) [13] enables routers to store different loop free paths to the closest locations of the named content in an ICN. It is scalable in context of signaling overhead and time required for getting correct routing information to named data. The researchers have utilized and analyzed the feasibility of the hash-routing methods in ICN [14]. They have contributed five hash-routing schemes to make use of in-network caching without demanding network routers to store state information for each content. This approach leads to significant increment in the cache hit ratio over on-path caching with less effect on the traffic dynamics of routes within the domain.

The above mentioned routing protocols for ICN designed for NDN architecture, emphasis on forwarding of interest packet to content producers. It does not focus on NDN's prominent benefits, such as multipath forwarding and in network caching. This actually gives a limit to the potential of NDN as well as advantages of its applications. In the proposed work of this paper, it is aimed at the exploiting the features of NDN architecture to provide content routing. It manly focuses on improving the user experience like content retrieval delay and managerial overhead reduction like network load. The proposed primarily exploits in-network caching feature of NDD architecture for realizing ICN technology.

III. Proposed work

A. System model

The objective of this work is to fetch content along fastest possible path so that latency could be minimized. The network model is assumed to have N nodes with enough storage space to accommodate caching of C number of contents. There are total M numbers of content sources, where the content is always available. Other than these producers of content, all nodes have cache capacity C. The size of content is equal and assumed to be U. The user in the network produces interest packet for a required content as per the norms of NDN architecture. Such content request is raised at the rate R. The content requests follow the Independent Reference Model (the IRM model). The working model of this paper assumes the Dijkstra's shortest path routing as a base to forward content requests in the network and proposed algorithm runs on top of it. In the network, all nodes have adopted policies related to cache management like LCE, LCD and uses replacement of cache like LRU. It also assumes that the content dissemination follows same route that the request packet has chosen, from requestor to cache location, but in reverse direction.

Primary concern of this research is to provide a routing strategy to run on top of the existing routing strategy thereby improving performance at network level. Targeted parameters are end-to-end latency, cache hit ratio and routing overhead. The proposed approach demands minimal modifications to existing routing solutions so that its deployment, testing as well as integration to other ICN prototypes become easy.

B. Operation model

The operation of the proposed mechanism starts with the demand of a content by a consumer. Whenever a consumer or client wants a particular piece of data, it produces an interest packet related to that piece of content. Our proposed protocol is used at this point to forward interest packets in the network. Once it has found out the node from where it can download the desired content, the request gets satisfied thereby matching the name of content mentioned in the request packet. Corresponding node sends the related content to the requestor. When data packet traverses from that node to the requestor, all nodes

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/ANTS47819.2019.9118123 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)

Algorithm 1: Functions associated with requester R

```
    = Requestor R's lookup table
      = Betweenness centrality table of content router R
N = Set of neighbours of R
Generation of interest message for content named i
Find     = Set of caches that holds content i in
Find C_RNi = set of caches that holds content i in T_j , jϵ
If (C    != Ø) then
              = set of shortest routes to each cache in
              = distance of each route in P
        Forward the interest packet to the cache with min{
        }
Else if (          != Ø) then
               = set of shortest routes to each cache in
               = distance of each route in P
        Forward the interest to the cache with min{          }
Else
     a) Compute the BC for each CR and Send it to its 1-
        hop neighbors.
     b) Build          for each CR named n_1,    , …n_n.
     c)        = { n_1—BC_n1---NH…..n_n---BC    --NH } ;
        n =  1 hop neighbors of CR in network,  NH means
        the next hop to reach associated CR.
     d) Forward the interest to the CR at the shortest
     e) CR will scan BC table of itself to find CR or     with
        max {BC    } and related NH.
     f)       =cache  of node     with maximum BC and
        holds the content i in
          If (C      != Ø) then
               CR will Send interest to that NH node
          Else
               Forward interest to content origin server.
          End if
End if
```

Fig. 1 Proposed algorithm for routing strategy in ICN

along the route may cache data based upon the policy related to caching.

In order to achieve this, each requestor maintains information in a lookup table. The said table stores data related to the node from where requestor has recently retrieved the content. Router maintains the lookup table in addition to the FIB table of specified by NDN architecture. FIB table is used to forward requests to cache. The existing routing protocol is responsible for filling entries in FIB and our proposed procedure runs on top of the base routing strategy. It supports this fact by introducing lookup table to find the caches that hold the content as well as from where the requested content is most likely to be present.

Every tuple in the lookup table comprises of the content name and name of the node/cache that holds the content. In order to preserve freshness, each row has a TTL value. TTL signifies the amount of time in the near future, until which a particular entry of the table may remain present inside the cache store. When the current system time becomes higher than the TTL time for a tuple, the entry of tuple us discarded. The computation of TTL value for a tuple is explained in later section of this paper. It is believed that if some user has retrieved a content in the recent past, then there is a possibility of residing the content in that location for quit sometime. So, in place of routing the interest packet towards origin server, it may be routed to the recent location to minimize the delay.

Algorithm in Fig. 1 depicts the proposed method for interest packet routing initiated by a consumer. Let us assume that a requester P generates an interest packet for content I. It first scans its own lookup table (say TR) to see the cache from where the requestor has recently downloaded the content. In case of multiple entries in the table, the interest packet will be forwarded towards the cache at nearest distance. If the requestor doesn't find the corresponding entry of cache for the requested content, it sends interest packet to its directly connected 1-hop neighbours. If neighbour requestor found the related entry, it replies to interest message with a list of cache locations from where the user can retrieve the needed data. Based upon that, requester sent request packet towards cache at the nearest location. If both previous searches are unsuccessful, the requestor sends the interest packet to one of its directly connected and nearest content router. The major contribution of our work is in this part of the process and the way our work differs from existing characteristic time routing [8]. In [8] the unsuccessful search is forwarded by the CTR towards origin server. But in our work case, each content router needs to calculate its betweenness centrality (BC) value and send the same to its 1-hop neighbors. Based upon the received BC values, each CR builds BC table that stores name of the CR, its BC and next hop. Whenever CR receives the interest packet by requestor after 2 unsuccessful lookup table searches, it first scans its own BC table to find which CR has highest BC value and what is the next hop in order to reach there. The interest packet is forwarded to that node through next hop node. The significance of BC parameter with the BC table structure is explained below.

In graph theory, one of the measures of centrality is betweenness centrality (BC). In terms of network topology, it is a centrality measure in a network depending on shortest paths. For each pair of nodes in a network, there exists at least one shortest route among the nodes such that either the number of links that the route goes through (for unweighted topology) or the addition of weights of links (for weighted topology) is minimal. The BC for each node is the number of these shortest routes that pass through the node. BC has a broader range of practical applications in computer network theory. It represents the degree of which CRs stand between one another. For example, if a CR has higher value of BC, it will have more control over the rest of network, because more data will be travelled through that CR. It implies that chances are high that it might get cached at that node which leads to a cache hit for the requested content.

$$BC(n) \qquad \frac{\sum_{p} \qquad _{pq(n)}}{} \tag{1}$$

Where σ_{pq} indicates total number of shortest routes from CR named p to CR named q and $\sigma_{pq}(n)$ is number of those routes that travel through CR named n. the BC of a node scales with the number of pairs of CRs as suggested by the summation indices. So, the computation may be rescaled by division of it with number of pairs of CRs,

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/ANTS47819.2019.9118123 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)

with no inclusion of n such that BC∈[0,1]. The division is done by (N-1)(N-2)/2 for undirected topology and (N-1)(N-2) for directed topology, where N denotes the number of CRs in the topology. This scaling signifies for the highest possible value, where one CR is passed by every single shortest route. The format of BC table with respect to a random node A1 is given in Table 1. This indicates that CR named A1 is directly connected to A2. And as only two CRs are present inside the network, BC will be 0, based on equation 1, as no intermediate CRs are there. It is assumed that A2 is the neighbour of node A1.

Table 1: Betweenness centrality table

CR Name	BC value	Next Hop
A2	0	A2

The purpose behind using BC as a measure of centrality is, it denotes an extent to which any node lies on the shortest route between other CRs. After receiving such information related to the node, the next hop router is selected. An interest packet is forwarded towards that node by referring FIB table entries. This actually increases the likelihood of content being fetched successfully from the CS of the content router that has high centrality value. So no need to go till content source to retrieve content. This will actually reduce the delay for content retrieval.

The primary challenge of the proposed algorithm is to determining an optimal TTL value for each entry in lookup table. As discussed earlier, a very small or large value of TTL leads to frequent purging of entries and probable cache miss respectively. In this paper, concept of probabilistic time is used to model the TTL parameter. There are numerous options available to compute this probabilistic time. The method used in this work is as follows. Let's assume that the probabilistic time is denoted by T_p and used to update the lookup table of the requester. Every caching node computes T_p for the content present in its store taking into consideration content passed through it and popularity of content. For a cached content request at time t, it also attaches the parameter TTL= t + T_p in the header of the data packet being sent back to the requestor. Once the content reached requestor, the requestor fetches the TTL from the packet header and modifies its own lookup table. The format for a lookup table for a sample node U3 is given in Table 2. This indicates U3 has accessed content c-1 from content router A1 and having related TTL value as TTL1.

Table 2: Lookup table structure

Content name	Source name	Probabilistic time
c-1	A1	TTL1

Algorithm 2: Function of cache

1. *Cache hit :*
2. Attach TTL to packet header which is being fetched.
3. Data retrieved along with the exact but reverse path.
4. *Cache miss :*
5. Refer BC table of CR and sent a request to CR that has max BC value
6. *Still cache miss:*
7. Send interest packet towards origin server.

Fig. 2 Function of cache

It is not guaranteed that content will be inevitably present inside cache to which the interest packet is directed. It may get deleted or unavailable in its expected location. So, if any cache miss occurs, the nearest connected CR refers to its BC table and forward the request towards a CR having highest BC value. It increases likelihood of cache hit. If still a cache miss occurs, the corresponding CR sends interest packet to origin server by referring FIB table as mentioned in above Algorithm 2 described by Fig. 2.

IV. SIMULATION SETUP AND RESULTS

The proposed algorithm is simulated in NS3 based ndnSIM simulator in order to exhibit and analyze the behavior. The base network scenario consists of 2 consumers, 6 content routers and 1 producer connected with a point to point links as depicted in Fig. 3. The scalability of the proposed scheme is evaluated by testing the performance by varying number of consumers and content routers.

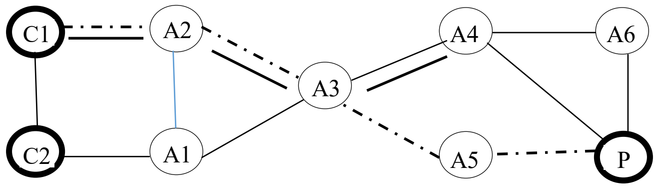

Fig. 3 Basic simulation scenario

It is assumed that request arrival rate R, content universe U, cache capacity C, Lookup table and BC table sizes LS and BS respectively. In the simulation values of these parameters are taken as U=500, R=20, LS=BS=300 respectively. Probabilistic time (T_p) is computed as per Che's approximation as mentioned in [12]. As shown in Fig. 3, c1 and c2 are 2 requestors, A_i where i=1 to 6 denotes content routers and P denotes producer with specified capacity. It has all the requested data within it. Each requestor maintains a lookup table and each CR maintains BC table with above specified sizes. Both data structures have been implemented as a multidimensional array. The base forwarding strategy and caching scheme used are Dijkstra's shortest path routing and LRU (max cache size=100) respectively. It is assumed that lookup table of c2 is empty and look up table of c1 has an entry corresponding to data item id2. The content router A4 has cached data item id1. The performance of the network is analysed when user c1 requests content id1. The BC values for A1, A2, A3, A4, A5, A6 are 0.1307, 0.1307, 0.3686, 0.095, 0.095 and 0 respectively for N=5 as per formula mentioned in Eqn. (1). Performance of the proposed scheme is compared with CTR algorithm [8].

Let us consider a case when c1 has sent interest to A2 for id1. Following proposed scheme, first C1 check its own and then neighbour's lookup table for an entry related to id1. But because of unsuccessful search it sends it to the nearest CR named A2. A2 will check its own CS and forward the interest to the CR which has the highest BC value, which means to A3. Again A3 will check its own CS and forwards it to the CR which has highest CR, which means to A4. Now A4 has already cached the item id1. So it will send the corresponding data packet to C1 and C1 will make corresponding entry of id1 in lookup

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/ANTS47819.2019.9118123 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)

table with its source and TTL value. So the path that interest packet follows is c1-A2-A3-A4 (the dark line in Fig. 3). For unsuccessful lookup table searches, the c1 will send interest towards the P, following shortest path route which is C1-A2-A3-A5-P mentioned (dotted line in Fig. 3). The path selected by CTR approach involves more nodes which in turn lead to high delay and overhead. The evaluated performance parameters for the proposed scheme against CTR scheme are content retrieval delay, cache hit ratio as well as an average packet overhead.

Fig. 4 shows relationship among data retrieval delay and distance (in hops) between user and data source. As the distance increases the time required to receive the corresponding data increases resulting higher delay in CTR. But the proposed scheme has less delay value compared to CTR as chances are of having the content in a node with high BC value is high. As the distance increases, there is 12-15% reduction in delay for d=5 and 6. For lower values of d, there is a 5-6% reduction in delay. The results have been examined for distance values from d=1 to 6.

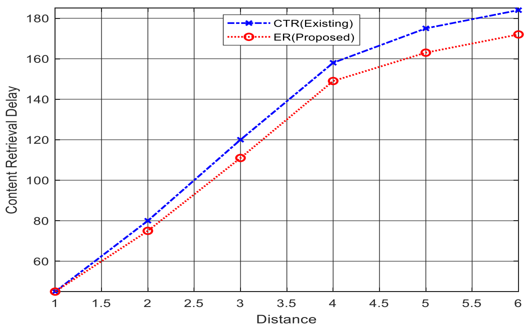

Fig. 4 Distance vs. Content retrieval delay

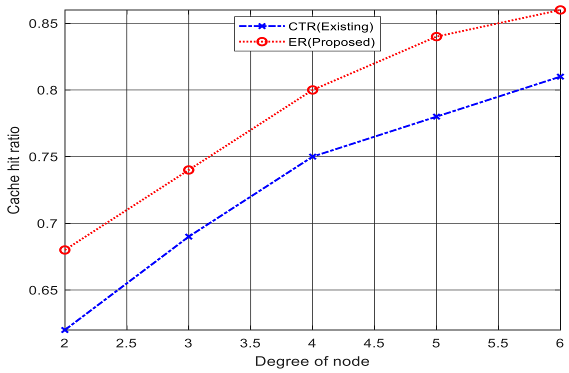

Fig. 5 Degree of node vs. Cache hit ratio

Fig. 5 shows how the degree of node impacts cache hit ratio for existing as well as proposed scheme. It signifies the rate of no of cache hits to the total number of lookups, scaled within the range of 0 to 1. When a network contains a number of nodes that have degree between 2 to 6, the BC value of that node increases as degree increases, which leads the inclusion of those nodes in the routing path and there will be a high likelihood of cache hits. As the degree of nodes increases, there is 5-6% of increment in the cache hit ratio of proposed scheme compare to existing routing scheme. These simulation results can be obtained by using available trace sources of ndnSIM classes. Fig. 6 shows the relationship between a number of nodes and the packet rate in kilobits/s for the amount of interest and data messages forwarded by a node of NDN. As the network becomes denser, it will have more nodes with high BC value with a high probability of having desired content being cached in its CS.

This ultimately reduces the required number of interest messages, destined to be sent towards a producer and because of which compare to CTR, an average kbits/s value will be reduced 10-12% in case of the proposed scheme for the number of content routers varying from n=3 to 21.

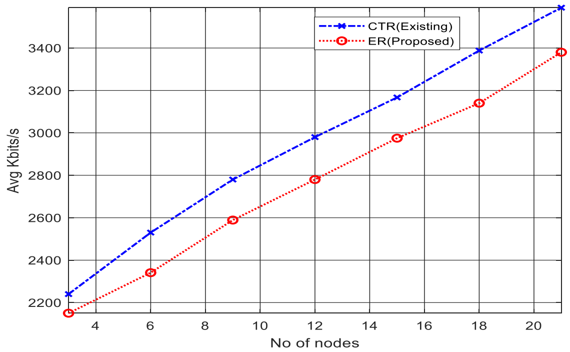

Fig. 6 No of nodes vs. Average packet rate

All results have been analyzed for request arrival rate 20 and simulation time of 20 seconds for the 2 consumers, 1 producer and varying number of CRs from 3 to 30. Now if we increase the number of users or requestors requesting for the content present in P, the avg kbits/s will be increased as there will be more number of interest packets and data packets flooded inside network. We have performed simulation for varying number of requestors from 2 to 10. We have analyzed that compare to existing CTR approach, proposed approach produce less amount of packet overhead inside network as per the observations mentioned in Fig. 7.

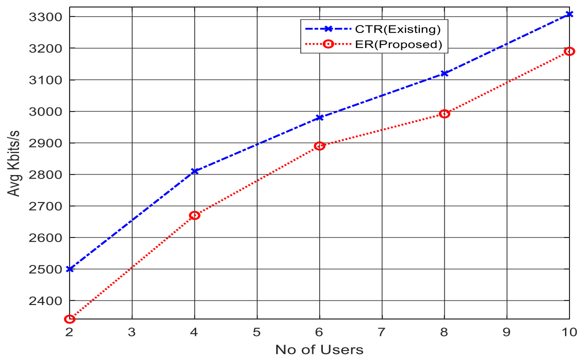

Fig. 7 Number of users vs. Avg Kbits/s

V. CONCLUSION

In this paper, a routing mechanism for ICN is proposed. The stated protocol executes on top of the Dijkstra's shortest path algorithm and work with available caching mechanisms. It utilizes the concept of likelihood time and betweenness centrality of a node in order to route the content requests towards the cache locations, where it can be found with maximum likelihood. The protocol is examined through extensive simulation and reveals that the proposed scheme had achieved superior performance.

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/ANTS47819.2019.9118123 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)

REFERENCES

[1] Aggarwal, Mayank, Kumar Nilay, and Kuldeep Yadav. "Survey of named data networks: future of the internet", *International Journal of Information Technology* , vol. 9.2 , pp. 197-207, 2017.

[2] I. U. Din, S. Hassan, M. K. Khan, M. Guizani, O. Ghazali, and A. Habbal, ''Caching in information-centric networking: Strategies, challenges, and future research directions'', *IEEE Commun. Surveys Tuts.*, vol. 20. 2, pp. 1443–1474, 2018.

[3] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. Polyzos, "A Survey of Information Centric Networking Research", *Communications Surveys Tutorials, IEEE*, vol. 16. 2, pp. 1024–1049, 2014.

[4] Lan Wang and A. K. M. Mahmudul Hoque, "OSPFN: An OSPF Based Routing Protocol for Named Data Networking", *NDN Technical Report, NDN-0003*, pp. 1–15, 2012.

[5] A. K. M. M. Hoque, S. O. Amin, A. Alyyan, B. Zhang L. Zhang, and L. Wang, "NLSR: Named-data Link State", *3rd ACM SIGCOMM workshop on Information-centric networking* , pp. 15-23, 2013.

[6] V. Lehman, A. Gawande, B. Zhang, L. Zhang, R. Aldecoa, D. Krioukov, and L. Wang, "An experimental investigation of hyperbolic routing with a smart forwarding plane in ndn", *In Quality of Service (IWQoS), IEEE/ACM 24th International Symposium*, pp. 1–10, 2016.

[7] I. Voitalov, R. Aldecoa, L. Wang, and D. Krioukov,"Geohyperbolic Routing and Addressing Schemes", *ACM SIGCOMM Comput. Commun. Rev.*, vol. 47. 3, pp. 11–18, Sep. 2017.

[8] Banerjee, B., Seetharam, A., Mukherjee, A., & Naskar, M. K., "Characteristic time routing in information centric networks", *Computer Networks*, vol.113, pp. 148–158, 2017.

[9] J. V. Torres, I. D. Alvarenga, R. Boutaba, and O. C. M. B. Duarte, "An autonomous and efficient controller-based routing scheme for networking Named-Data mobility", *Comput. Commun*., vol. 103, pp. 94–103, 2017.

[10] J. Torres, L. Ferraz, and O. Duarte, "CRoS-NDN: Controller-based routing strategy for named data networking", *Technical report, Electrical Engineering Program, COPPE/UFRJ*, 2014.

[11] C. Ghasemi, H. Yousefi, K. G. Shin, and B. Zhang. ,"MUCA: New Routing for Named Data Networking", *In IFIP Networking.* pp. 289–297, 2018.

[12] Imai, S.Leibnitz, K.; Murata, M.," Statistical approximation of efficient caching for one-timers", *IEEE Transaction on Network and service management*, vol. 12.4, pp. 595-604, 2015.

[13] J. J. Garcia, Luna-Aceves, "Name-based content routing in information centric networks using distance information", *1st ACM International Conference on Information-centric Networking, ICN*, pp. 7–16, 2014.

[14] L. Saino, I. Psaras, and G. Pavlou, "Hash-routing schemes for information centric networking", *ACM SIGCOMM ICN Workshop*, pp. 27–32, 2013.

[15] Beakal Gizachew and Assefa Oznur Ozkasap, "A survey of energy efficiency in SDN: Software-based methods and optimization models", Journal of Network and Computer Applications, vol. 137(1), pp.127-143, 2019.

[16] Mani Shekhar Gupta, Krishan Kumar, "Progression on spectrum sensing for *cognitive radio networks*: A *survey*, classification, challenges and future research issues", Journal of Network and Computer Applications, vol. 143, pp. 47-76, 2019.

[17]Nitul Dutta, HKD Sarma and Z. Polkowski, "Cluster based routing in cognitive radio adhoc networks: reconnoitering SINR and ETT impact on clustering", Com. Com., (Elsevier), pp. 10-20, vol. 115, 2018.

[18] Nitul Dutta, HKD Sarma, "A probability based stable routing for cognitive radio adhoc networks", Wire. Net., (Springer), vol. 23(1), pp. 65-78, 2017.