Indian Journal of Science and Technology, Vol 9(37), DOI: 10.17485/ijst/2016/v9i37/100323, October 2016

Selective Placement of Caches for Hash-Based Off-Path Caching in ICN

Anshuman Kalla* and Sudhir Kumar Sharma

Department of Electronics and Communication Engineering, Jaipur National University, Jaipur - 302017, Rajasthan, India; kallajnu@gmail.com, sudhir.732000@gmail.com

Abstract

In-network caching has evidently emerged as an indispensable core functionality of ICN. Of the various types of caching techniques, off-path caching turns out to be a potential technique. With a view to improve the performance of hash-based off-path caching, the aim of the paper is to ensure profitable, economic and selective placement of caches such that the final cache allocation tends to reduce simultaneously average retrieval delay and maximum internal link-stress. To achieve the same, we propose simple yet elegant heuristic algorithm to strategically identify and exclude the bad node-positions for placement of caches. More specifically, for a given topology, the algorithm computes cost (in terms of delay) for all the nodes and prepares a cost based descending ordered list of nodes. Further, iteratively, increasing number of nodes, in sequence, from the cost-based ordered list of nodes are debarred from caching thereby resulting in family of solutions. A solution represents a selective cache allocation map corresponding to the number of debarred nodes. Moreover, every iterative solution reduces average retrieval delay and seeks opportunistic reduction in maximum internal link-stress. Comparing the solutions graphically would enable network operators to finally select (as per desire) one cache allocation map for a given network topology for hash-based off-path caching. In a way, the selected cache allocation map for the considered topology implies that it is better to place the caches at the core of the network as compared to edge.

Keywords: Content Centric Networking, In-Network Caching, Information Centric Networking, Off-Path Caching, Selective Cache Placement

1. Introduction

Research based resurgence in the field of networking seems to have culminated in form of Information Centric Networking (ICN) since it has the capability to conciliate in a unified manner all the existing and anticipated issues. ICN draws its strength from its well-known salient and integral features¹. In-network caching is one of those features and is expected to play a vital role in ameliorating the network performance and user's experience. Various advantages of in-network caching and performance metrics used to evaluate its efficacy are discussed in². Content

Centric Networking (CCN) architecture and its related terminology have been considered as fundamental base for rest of the paper.

Up till now, numerous types of caching techniques have been proposed and worked upon by researchers all around, in context of ICN. Off-path caching is one of the types of in-network caching that has preliminary shown better results as compared to edge caching³ and on-path⁴⁻⁹ caching and thus seems to have the required potential. Nevertheless, one of the downsides of off-path caching is that it trades-off (i.e. increases) intra-domain link load to achieve better performance³⁻⁶. One of the practical and

^{*}Author for correspondence

popular ways of implementing off-path caching involves the use of hash function and is referred in this paper as Hash-based off-path caching³⁻¹⁰.

In pervasive* hash-based off-path caching with LRU as replacement policy, the cache hit ratio (consequently the bandwidth utilization of the external links and server load) primarily depends on the factors like overall network cache size, cache diversity and popularity profile of content requests3-5. In addition to above mentioned factors, (the metrics) average retrieval delay and internal link-stress are dependent on the topology as well. Hence with this background, the questions that we pose is: Does there exists a selective (thus economic) cache allocation# map which could be conducive in achieving smaller average retrieval delay and at the same time lower maximum intradomain link-stress, while keeping the other relevant metrics (cache hit ratio, server-load and external link utilization) unaltered as compared to pervasive cache allocation? If so, what could be a simple way to find such a selective placement* of uniform sized caches for hash-based off-path caching technique?

We attempt to answer these questions by proposing a simple heuristic algorithm that aims to find a selective cache allocation map by excluding the bad node-positions for placement of caches. So, in a way, we turn around the question and ask another question: Is it possible to reduce both the average retrieval delay and the maximum internal link-stress by debarring some of the nodes from caching (while keeping the network cache budget fixed) giving rise to a selective cache allocation map? The rationale is that there might be few nodes in a network which, if selected for caching, cause adverse effect on the average retrieval delay and internal link-stress. The algorithm ensures that the aggregate network cache budget, for selective placement of caches, is same as that used for pervasive placement of caches (i.e. same before & after exclusion of caches).

Besides the reduction in average retrieval delay and opportunistic lowering of maximum internal link-stress there is yet another advantage, though latent, of the proposed selective cache allocation for hash-based off-path caching. That is, without sacrificing the performance, the algorithm decreases the number of nodes involved for innetwork caching thereby results in significant monetary savings11 (CAPEX and OPEX). Though the problem of cache allocation has been worked upon by researchers¹²⁻¹⁴ but to the best of our knowledge, selective placement of caches in context of hash-based off-path caching has not been explored so far. Thus the present work is the first attempt in this direction.

Rest of the paper is organized as follow. Next, section 2 discusses the state-of-art of in-network caching in ICN. Section 3 introduces the selective cache allocation algorithm for hash-based off-path caching. Following that, section 4 deals with evaluation of the proposed algorithm and discusses the results obtained. Finally, the section 5 concludes the work done.

2. State-of-Art

Developments and some of the work carried out concerning in-network caching are summarized below:

2.1 Hybrid Caching

The paper¹⁵ coupled on-path caching with conditional offpath caching. In case an intermediate node does not holds a cached copy but has witnessed transit of the requested content within last predefined time window then instead of normal (on-path) forwarding of interest packet, the node deflects it off-the-path towards closer replica. Authors in paper¹⁶ introduced cooperative modulo caching strategy for effective deployment of time-shifted video streaming service in CCN. Every node is being assigned a unique integer label (small than a fixed value *k*). Over the retrieval path, if modulo k of the chunk's sequence number is equal to the label of an encountered node then the traversing content is being cached at that node. In paper¹⁷, ICN's content-routers are allowed to mingle with regular IP routers. Authors proposed use of bloom filter to share cache digest which allows content routers to realize caching-based indexed structure. In case the requested

^{*}Cache placement or allocation means configuring the nodes with content storage capacity.

^{*}Pervasive means all the nodes are enabled with in-network caching capability

content is not cached at an intermediate node then based on the indexed structure the node generates auxiliary scanning requests which are sent out (off-the-path) to tap neighboring content-routers. It aims to reduce retrieval latency and balance load among the links.

The technique proposed in paper¹⁸ allows every node to know the contents' popularity ranking (estimated distributively) and to compute a node-cost list based on received request rate and latency from other routers. Any node with this two-fold information figures out the contents it is suppose to cache and the contents which collaborating routers are suppose to cache (i.e. contentto-cache mapping). Yet another hybrid caching is intra AS cooperative caching technique¹⁹ that blends on-path caching with restrictive one-hop off-path caching. If the requested content is not found at an intermediate node then before forwarding the request upstream towards the server, it consults its immediate (one-hop) neighbors.

Authors in work²⁰ put-forward a hybrid cache coordination scheme and suggest to divide a network into two parts. First part comprises of single core area and deploys off-path caching to ameliorate cache diversity. Whereas, the second part represents collection of several edge areas that adapt on-path caching (thereby allows cache redundancy) to enhance user's experience by caching popular contents close to the users. With similar intent to improve user's experience in terms of retrieval latency, the paper³ clubbed off-path caching with edge caching. Authors proposed to dedicate small portion of content storage at ingress nodes for edge caching and remaining for regular off-path caching. Further, with help of reference based (local) popularity estimation and pre-fetching of popular contents at edges, it has been demonstrated that retrieval delay for the few top-most popular contents could be reduced by sacrificing marginally overall network performance. The paper²¹ proposed cooperative caching where caching at edge routers is being assisted by off-path caches placed strategically. Further, benefit for every content is computed based on distance between requester & server and access frequency. If the benefit of fetched content is better than any of the cached contents at the edge router (present on the retrieval path) then the fetched content is cached by replacing the already residing content with lower benefit. The replaced content is referred for off-path caching.

2.2 Off-Path Caching

The paper4 developed theoretical framework for off-path caching to minimize traffic over expensive inter-domain links and load on origin server. Further, authors introduced CACH (Cache All Content by Hashing) heuristic as a practically implementable off-path caching technique. CACH indiscriminately allows caching of all the contents (popular or non-popular) by placing them at designated off-path caches determined by hashing content name. The job of retaining the popular contents is left to LRU replacement policy. Authors in paper⁵ identified three different instantiations of hash routing for off-path caching based on the retrieval path being followed by content packets: Symmetric (i.e. data packets track back exactly the same path traversed by the interest packets), Asymmetric (distinct paths tracked by interest and data packets) and Multicast (multiple retrieval paths concurrently being followed by data packets). The paper⁶ motivates to divide a network into k-clusters in a way that at maximum one copy of a given content could be cached per cluster. Thus the rationale is to seek a sub-optimal off-path caching which endures (restrictive) cache duplication thereby trading-off cache diversity (and thus cache hit rate) in order to gain reduction in intra-domain link bandwidth usage.

For a set of inter-connected ASes, the paper⁷ proposed an implicit coordinated off-path caching technique. To allow every AS to gather knowledge of its neighboring ASes along with the content range (i.e. sector of entire namespace) they are willing to cache, authors advocate use of Pathlet routing protocol. Based on that, for an incoming interest packet, a path towards the server is selected in such a way that the interested AS falls that path. Moreover, within an AS, hash function is used to independently deflect the request towards the designated router (as in work4).

Unlike4,16 where content catalogue partitioning is achieved by hashing and modulo techniques respectively, CPHR⁸ proposed two-level partitioning with the aim to balance workload among caches. At the first level, content-catalogue is partitioned in subsets based on egress nodes (used to receive a given content from outside). As a result one partition per egress node is formed. At second level, every subset is further divided into k-partitions to which the hash values are mapped. The paper¹⁰ tackled the issue of inconsistent hashing (which leads to unbalanced caches when node failure or node insertion occurs) by (i) clustering all the nodes in two different groups and (ii) engaging the technique of consistent hashing along with concept of virtual routers.

The paper⁹ introduced CoMon as centrally coordinated off-path caching. The work seems to follow similar philosophy that of paper¹⁸ however is carried out in centralized manner. One central controller, after receiving reports (containing local request patterns) from all the appointed coordinators, compiles an aggregate popularity profile. Further based on betweeness centrality of

nodes, the controller maps the estimated top most popular contents to the nodes. Nodes then ensure the caching of assigned contents.

3. Heuristic Algorithm - Placement of Caches

The proposed heuristic achieves its goal of improving performance of hash-based off-path caching by computing cost for each node in a given network topology. Following equation is being used to compute cost C_r for every node r in a network topology comprised of \mathcal{V} set of nodes.

$$C_{r} = \sum_{(i \in I)} \{ \mathcal{H}^{*} d_{ir} + [(1 - \mathcal{H})^{*} (d_{ir} + d_{re})] \} \quad \forall r \in \mathcal{V}$$

$$\tag{1}$$

Algorithm 1 Selective placement of caches for hash-based off-path caching

Input: Network topology (i.e. set of nodes V with link latencies), hit ratio (\mathcal{H}) of pervasive hash-based off-path caching and parameters from table 1

Output: A solution representing selective cache allocation map for hash-based off-path caching

Ensure: Aggregate network cache budget is same for all iterations

```
create a list of all the nodes in network i.e. L \leftarrow \mathcal{V}
1.
2.
        for all the nodes in L do
3.
            compute cost C_r as per equation (1)
4.
        end for
5.
        create an ordered list L_c \leftarrow nodes in L sorted in descending order of cost C_r
6.
        for i = 1: (length of L_c)
7.
          \mathcal{N}_i \leftarrow L_c [1 to i] // top i costly nodes to be debarred
8.
             allocate uniform cache size to V \setminus N_i nodes selected for caching
9.
             run hash-based off-path caching with V \setminus N_i caches
10.
                   - compute hit ratio \mathcal{H}_i
                   compute cache diversity \mathcal{D}_i
11.
                   compute average retrieval delay \mathcal{A}_i
12.
                 compute maximum internal link-stress S_i
13.
         \subset S_i^F \leftarrow \{\mathcal{N}_i, \ \mathcal{H}_i, \ \mathcal{D}_i, \ \mathcal{A}_i, \ S_i\} // creating family of solutions
14.
        end for
15.
        compare S_i^F for different values of i
16.
17.
        select a desired off-path cache allocation pattern as per requirement
```

Since uniform rate of request arrival at ingress nodes is assumed thus above equation is not weighted by rate of interest request at ingress nodes. The equation makes use of two factors; (i) internal delay dependant on topology and (ii) hit ratio (\mathcal{H}) of pervasive hash-based off-path caching. Further, durepresents delay encountered to reach destination y from source x using the shortest part. Set I denotes set of ingress nodes that receive content requests in form of interests. $e \in E$ denotes egress node that is connected to server and falls over the shortest path to server.

Based on the computed cost as per equation 1, algorithm 1 sorts the nodes in the descending order (most costly to the least costly) and an ordered list of nodes (L_c) is prepared. Next, the algorithm iteratively appoints increasing number of nodes $(\mathcal{N}_i \subset \mathcal{V})$, from the costbased ordered list of nodes (L_c) and debar them from caching. Such debarred nodes are absolutely transparent, in terms of caching, for the named-content packets traversing them. Remaining nodes ($V \setminus N$, is complementary set) forms an instance of selective cache allocation map. Next, fixed aggregate network cache budget is equally distributed among all nodes $(V \setminus N)$ permitted for innetwork caching. In other words, uniform sized cache per node is allocated. Hash-based off-path caching is then carried out with the selected nodes.

Note, for performance comparison, the aggregate network cache size is kept same for all the iterations and is equal to that being used for pervasive hash-based off-path caching. Iterations result in family of solutions (S_i^F) corresponding to the *i* nodes debarred from caching. A solution represents a selective cache allocation map and is characterized by various performance metrics; hit ratio (\mathcal{H}_{i}) , cache diversity (\mathcal{D}_{i}) , average retrieval delay (\mathcal{A}_{i}) and maximum internal link-stress (S_i). Comparing these solutions could enable network operator to finally select one solution (i.e. cache location map) as per requirement.

4. Evaluation

In order to evaluate efficacy of the proposed algorithm, we developed a discrete event-driven simulator. Exodus US, one of the real PoP level topologies from Rocketfuel²², has been used for evaluation. Table 1 shows the relevant parameters and assumptions that were taken into account for carrying out simulations. Some of those parameters have been inspired from papers^{3,4}. About 8% of nodes are randomly appointed as egress nodes and are connected to origin server. Further, approx 55% of nodes are randomly selected as ingress nodes and are connected to clients i.e. users. Aggregate network cache size is chosen to be 10% of content population.

At this point, there are three important things to note. First, for evaluation purpose only link propagation delay has been taken into consideration. Others factors like queuing delay, packet loss, cross traffic etc. have not been incorporated for preliminary evaluation. Second, for rest of the paper, iteration specifies a repetitive process of debarring increasing number of nodes from caching whereas a run indicates an act of deploying hash-based off-path caching with a selective cache allocation map. As per table 1, 20 runs per iteration have been carried out during the simulations. Third, since depending on the hash function being leveraged the one-to-one mapping (i.e. placement of content to a designated off-path node) could be entirely different, thus for better evaluation content-to-cache mapping is being changed for every run. This explains the reason behind multiple runs per iteration.

4.1 Performance Metrics

Average Retrieval Delay: The metric denotes retrieval delay encountered on an average to fetch a requested content irrespective of source-of-request and point-ofretrieval. It is an indicative of Quality-of-Experience (QoE) perceived by users. Smaller the delay better is the user's QoE.

Link Stress: For every intra-domain link, the metric indicates the volume of the traffic passed through it. Further, the value of link stress for the most heavily loaded link is termed as maximum internal link stress. In this paper, maximum link stress has been focused as it enables lucid comparison of caching techniques.

Hit Ratio: It is ratio of the requests being satisfied by contents residing at in-network caches to the total requests received. It signifies efficiency of an employed caching technique. Higher is the hit ratio, lower is the traffic delegated over the expensive external links and lesser is the load on origin server.

Unique Contents Cached: The metric specifies total number of distinct contents being cached by all the in-

 Table 1.
 Parameters for Evaluation

Parameters Values		
Total number of nodes	79	
Number of server	one	
Number of egress nodes (connected to server)	6	
Number of ingress nodes (connected to users)	44	
Content population	79000	
Aggregate network cache size	7900	
Cache size per node	100	
Popularity distribution of content requests	Zipf $(\alpha = 0.8)$	
Delay to reach origin server from egress nodes	100 ms	
Cache size per node (Selected for caching)	Uniform	
Rate of request arrival at ingress nodes	Uniform	
Content size	Homogeneous	
Cache replacement policy	LRU	
Number of iterations carried out for the selected topology	39	
Number of runs per iteration of algorithm 1	20	
Number of requests simulated per run (i.e. after launch of hash-based off-path caching)	500,000	
Routing	Shortest Path	
Links between users and ingress nodes	Bi-directional	
Network regime	Congestions Free	

network caches collectively. More is the unique contents cached better is the cache diversity.

4.2 Results and Discussion

Initially, we simulated pervasive hash-based off-path caching (i.e. with no node debarred) and recorded its performance. The value of hit ratio (\mathcal{H}) turned out to be 0.4637 and was used as an input to the proposed algorithm (for cost computation of nodes in the topology). Next, as per the parametric values delineated in Table

1, the algorithm – 1 is applied to Exodus US topology. Figure 1 exhibits the reduction achieved in average retrieval delay with respect to the increasing number of nodes being debarred from the cost-based ordered list of nodes (L_c). Since 20 runs per iteration are performed thus mean values (along with standard deviations) of the average retrieval delay are plotted. Corresponding to the same number of debarred nodes, figure 2 reveals the maximum link stress observed. Comparing the two figures, it could be concluded that debarring first 15 nodes

(from cost-based ordered list) leads to a selective cache allocation map that results in maximum simultaneous reduction in average retrieval delay and maximum link stress without deteriorating the values of other performance metrics. Thus this selective cache allocation map corresponding to 15 debarred nodes is chosen for further study. Observing carefully the selected cache allocation map reveals that 13 out of 15 debarred nodes are the edge nodes (that are connected directly to the clients). This implies that it is beneficial to place the caches at the core of the network in contrast to the network edge.

Figure 3, depicts the content-wise average retrieval delay and content-wise hit ratio for the chosen selective cache allocation and pervasive cache allocation in context of hash-based off-path caching. Plotted are the mean values of the two metrics computed over 20 runs. It is apparent that retrieval delay for selective hash-based offpath caching is lower as compared to pervasive caching.

Table 2 evinces the gain in performance achieved by the chosen selective cache allocation as compared to pervasive cache allocation for hash-based off-path caching. The proposed algorithm is able to achieve 6.65% and 5.35% reduction in average retrieval delay and maximum internal link stress respectively. Quantitatively the gain might not be high but qualitatively the proposed technique achieves the same at no trade-off. In other words,

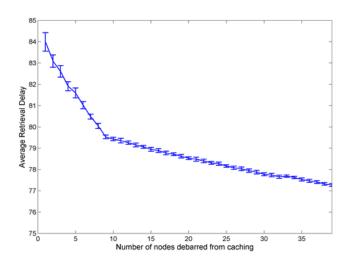


Figure 1. Average retrieval delay for selective hash-based off-path caching.

the beauty of the selective cache allocation is that the gain comes at no degradation in performance i.e. without sacrificing any other relevant performance metrics. In addition, since 15 nodes (out of 79) are debarred from caching (corresponding to the chosen selective caching allocation map) thus these nodes are not to be upgraded with the functionalities of ICN which in-turn results in achieving lower initial and operational cost compared to pervasive cache allocation.

5. Conclusion

Selective cache allocation has been explored in context to hash-based off-path caching. In this regard, a simple heuristic algorithm has developed that aims to reduce the average retrieval delay and simultaneously seeks opportunistic decrement in maximum internal link stress. Notably, the algorithm achieves its aims without deteriorating other relevant performance metrics as compared to pervasive hash-based off-path caching. The results confirm that selective placement of caches for hash-based off-path caching has potential to improve the performance in-network caching and to cutback the initial & operational cost involved in upgrading the nodes with innetwork caching capacities.

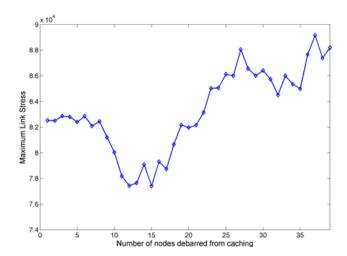
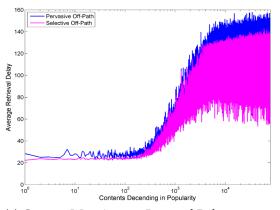
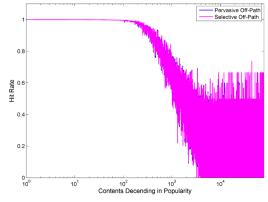


Figure 2. Maximum internal link stress for selective hashbased off-path caching.





(a) Content-Wise Average Retrieval Delay

(b) Content-Wise Hit Ratio

Figure 3. Performance comparison of hash-based off-path caching for Exodus US Topology.

Table 2. Performance comparison of pervasive and selective cache allocation for Exodus US Topology

Type of Cache Allocation	Hit Ratio	Average Retrieval Delay	Cache Diversity	Maximum Link Stress
Pervasive	0.4635	84.5661	7900	81784
Selective	0.4638	78.9465	7900	77410

6. References

- Zhang L, Estrin D, Burke J, Jacobson V, Thornton JD, Smetters DK, Zhang B, Tsudik G, Massey D, Papadopoulos C, Abdelzaher T. Named data networking (ndn) project. Tech. Report NDN-0001, Xerox Palo Alto Research Center-PARC. 2010 Oct 31.
- Kalla A, Sharma SK. A constructive review of in-network caching: A core functionality of ICN. 2016 International Conference on Computing Communication and Automation (ICCCA) IEEE. 2016 Apr 29.
- 3. Kalla A, Sharma S. Exploring off-path caching with edge caching in Information Centric Networking. 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), IEEE. 2016 Mar 11.p. 630–5.
- 4. Saucez D, Kalla A, Barakat C, Turletti T. Minimizing bandwidth on peering links with deflection in named data

- networking. INRIA Sophia Antipolis Mediterranee. Tech Rep. 2012 Mar.[Online]. Available from: http://hal.inria.fr/hal-00684453
- Saino L, Psaras I, Pavlou G. Hash-routing schemes for information centric networking. Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking, ACM. 2013 Aug 12; 27–32.
- Li C, Okamura K. Cluster-based In-networking Caching for Content-Centric Networking. International Journal of Computer Science and Network Security (IJCSNS). 2014 Nov 1; 14(11):1.
- Saha S, Lukyanenko A, Yla-Jaaski A. Cooperative caching through routing control in information-centric networks. In INFOCOM, 2013 Proceedings IEEE. 2013 Apr 14; 100– 4.
- 8. Wang S, Bi J, Wu J. Collaborative caching based on hash-routing for information-centric networking. ACM

- SIGCOMM Computer Communication Review, ACM. 2013 Aug 12; 43(4):535–6.
- 9. Salah H, Strufe T. CoMon: An architecture for coordinated caching and cache-aware routing in CCN. In2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC). 2015 Jan 9. p. 663–70.
- Thar K, Ullah S, Hong CS. Consistent hashing based cooperative caching and forwarding in content centric network. 2014 16th Asia-Pacific Network Operations and Management Symposium (APNOMS), IEEE. 2014 Sep 17; 1–4.
- 11. Perino D, Varvello M. A reality check for content centric networking. In Proceedings of the ACM SIGCOMM workshop on Information-centric networking. 2011 Aug 19; 44–9.
- Wang Y, Li Z, Tyson G, Uhlig S, Xie G. Optimal cache allocation for content-centric networking. In 2013 21st IEEE International Conference on Network Protocols (ICNP). 2013 Oct 7. p. 1–10.
- 13. Araldo A, Mangili M, Martignon F, Rossi D. Cost-aware caching: optimizing cache provisioning and object placement in ICN. 2014 IEEE Global Communications Conference 2014 Dec 8. p. 1108–13.
- Azimdoost B, Farhadi G, Abani N, Ito A. Optimal in-network cache allocation and content placement. In 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE. 2015 Apr 26. p. 263–8.
- 15. Rosensweig EJ, Kurose J. Breadcrumbs: Efficient, best-effort content location in cache networks. INFOCOM 2009, IEEE. 2009 Apr 19; 2631–5.

- Li Z, Simon G. Time-shifted tv in content centric networks: The case for cooperative in-network caching. 2011
 IEEE International Conference on Communications (ICC),
 IEEE. 2011 Jun 5. p. 1–6.
- 17. Lee M, Cho K, Park K, Kwon T, Choi Y. Scan: Scalable content routing for content-aware networking. In2011 IEEE International Conference on Communications (ICC), IEEE. 2011 Jun 5. p. 1–5.
- Guo S, Xie H, Shi G. Collaborative forwarding and caching in content centric networks. InInternational Conference on Research in Networking, Springer Berlin Heidelberg. 2012 May 21. p. 41–55.
- Wang JM, Zhang J, Bensaou B. Intra-AS cooperative caching for content-centric networks. In Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking, ACM. 2013 Aug 12; 61–6.
- Zhang G, Wang X, Gao Q, Liu Z. A Hybrid ICN Cache Coordination Scheme Based on Role Division between Cache Nodes. In 2015 IEEE Global Communications Conference (GLOBECOM), IEEE. 2015 Dec 6. p. 1–6.
- Rath HK, Panigrahi B, Simha A. On Cooperative On-Path and Off-Path Caching Policy for Information Centric Networks (ICN). In 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), IEEE. 2016 Mar 23. p. 842–9.
- 22. Spring N, Mahajan R, Wetherall D. Measuring ISP topologies with Rocketfuel. ACM SIGCOMM Computer Communication Review. 2002 Oct 1; 32(4):133–45.