Towards On-Path Caching Alternatives in Information-Centric Networks

Andriana Ioannou
School of Computer Science and Statistics
Trinity College
Dublin 2, Ireland
Email: ioannoa@scss.tcd.ie

Stefan Weber
School of Computer Science and Statistics
Trinity College
Dublin 2, Ireland
Email: Stefan.Weber@scss.tcd.ie

***Abstract*—Information-Centric Networking (ICN), an alternative to the current Internet architecture, focuses on the distribution and retrieval of content instead of the transfer of information between specific endpoints. Approaches to ICN employ caches in the network to eliminate the transfer of information over lengthy communication paths from a source to consumers.**

The contribution of this paper lies in the placement of copies in on-path in-network caching. Our goal is to investigate the suitability of a probabilistic algorithm, *Prob-PD*, based on two variables, the content's popularity rates and the distance ratio of each node from the source, with regard to caching performance, i.e. cache hit rates, cache replacement rates and content delivery times. Towards this goal, we present an initial comparison of simulation results of the proposed caching mechanism and published alternatives showing significant gains of the algorithm.

***Index Terms*—Network Distributed Architectures; Future Internet; Information-Centric Networks; Caching technologies; Content replication; On-path caching**

I. Introduction

Information-Centric Networking (ICN), an alternative to the end-to-end communication paradigm, focuses on content distribution and retrieval; content sources distribute their content by publishing it to a content notification service, i.e. a name resolution service or a name-based routing service, while clients retrieve content by subscribing to it. ICN architectures identify content resources, such as web pages or parts of a content resource, chunks or packets, using a content identifier. Content identifiers should involve no information that would bind the content to a specific location [4]. If this constraint is met, the content can be freely cached. Approaches to caching can be categorized into *off-path caching* and *on-path caching* with regard to the location of caches [22].

Off-path caching aims to replicate content within a network regardless of the forwarding path. Off-path caching is usually centralized and involves a great amount of information collected and advertised, in a content notification service. The ICN off-path caching problem is equivalent to the replication problem defined in Content Delivery Networks (CDNs) and web proxies [9], [22]. On-path caching on the other hand, is integrated to the architecture itself, i.e. the caching decision is limited to the content propagated along the delivery path and to the nodes on the delivery path, caching is accomplished at the network layer, thus, being independent of the application but bounded by the on-line speed requirements of the delivery process, where the overhead of monitoring, collection of statistical information or advertisement of the cached content in a content notification service may not be acceptable or feasible. Finally, on-path caching does not follow a specific topology. Both caching approaches can be applied either separately or as a combination. The benefits of on-path caching against the off-path caching are still under investigation [9], [10].

In this paper, we focus on the efficiency of caching mechanisms for the on-path caching problem identified in the area of ICN. Towards this goal we propose a probabilistic algorithm, *Prob-PD*, based on two variables, the content's popularity ratio *(P)*, and the distance ratio of each node from the source *(D)*, which we compare against the alternatives via simulations. A detailed description of this work can be found in [11].

The remainder of this paper is structured as follows. In section II, we describe the related work in on-path caching and conclude to the most efficient algorithms. In Section III, we describe the Prob-PD algorithm. In Section IV, we present the simulation model and the evaluation results of the Prob-PD algorithm against the most efficient alternatives concluded in Section II, as well as indications for future work. We close the paper with Section V, dedicated to the conclusions.

II. Related Work

In this section, a summary of the proposed on-path caching approaches and their evaluation results is provided, with regard to a number of performance metrics explained in Table I.

A variety of the proposed on-path caching algorithms are based on a probability p, according to which, a node n on the delivery path decides to cache a content. Probabilistic algorithms can be categorized depending on the way that probability p is calculated; based on a fixed value [3], [7], *FIX(p)* or based on a mathematical formula [17]. The CE^2 algorithm proposed by Jacobson et al. [12], is a FIX(p) algorithm with p=1. Rossi et al. [18] have tested the FIX(p) algorithms against the *LCD* approach, resulting in higher cache hit rates. The LCD approach caches the requested content one hop closer to the client each time that a content request arrives.

Psarras et al. [17] have suggested a probabilistic caching algorithm called *ProbCache*, composed by the *TimesIn* factor and the *CacheWeight* factor. The TimesIn factor indicates the

978-1-4799-3780-6/14/$31.00 ©2014 IEEE

TABLE I: Description of the evaluation metrics in section II.

Evaluation Metric	Description
Server hit rates	No. of content requests satisfied by a server
Cache hit rates	No. of content requests satisfied by a cache
Eviction rates	No. of cache replacements ocurred in a cache
Absorption times	Tot. time that a content stays cached in the system
Hop count rates	No. of hops that a content request travels
Download times	Tot. time to retrieve a content

capacity of the remaining nodes along the delivery path, favoring contents that travel from further away. The CacheWeight factor acts as a counter-balance to this unfairness. Psarras et al. have evaluated the ProbCache algorithm against the FIX(p), CE^2 and LCD algorithms, indicating significant gains in terms of server hit rates, hop count rates and eviction rates.

Sourlas et al. [20] have proposed an on-path caching algorithm called *LeafNode*. According to this approach, content is cached at the last node of the delivery path. Sourlas et al. have evaluate the LeafNode algorithm against the CE^2 approach, resulting in higher hop count rates and lower absorption times.

Rossi et al. [19] have examined the suitability of a number of graph-based algorithms, i.e. *Degree Centrality (DC)*, *Betweenness Centrality (BC)*, *Closeness Centrality (CC)*, *Graph Centrality (GC)*, *Eccentricity Centrality (EC)* and *Stress Centrality(SC)* for determining the size of the caches. Based on their evaluation, the DC approach, which indicates the number of edges on a node, is the most efficient graph-based algorithm in terms of server hit rates and hop count rates.

Concluding this section, three approaches, the FIX(0.90), DC and ProbCache, seem to outperform the rest of the alternatives. One of the main contributions of this work is the evaluation of these algorithms against each other. Based on this comparison and the previous ones performed by the research community, we expect to determine the nature of the caching system that would be more beneficial for an ICN architecture.

III. Probabilistic-PD Algorithm

Based, on the information summarized in section II, one can observe the absence of the content popularity as a criterion to the caching decision. Content popularity is an important factor, able to affect the performance of a caching algorithm and the network as a whole [6], [18]. Therefore, it should be taken into account. Content popularity has been applied on a number of cache replacement policies and replication algorithms defined in the areas of web proxies and CDNs, e.g. [13], [15].

Towards this direction and inspired by the Local Greedy algorithm, proposed by Kangasharju et al. [15] for content replication on CDNs, we propose a probabilistic algorithm, called *Prob-PD*, consists of two factors, the content's popularity ratio *(P)*, observed on a node and the distance ratio between the same node and the source serving the content *(D)*. At this point, one should make a decision regarding the way that content popularity is calculated. Based on this criterion, content popularity may be divided into *static-content popularity* and *dynamic-content popularity*.

Static-content popularity approaches require the definition of a threshold *h*. Contents with a number of requests higher than *h* are considered to be popular while contents with a number of requests lower than *h* are considered to be unpopular [8], [14]. Unpopular contents are excluded from the caching decision. Due to the volatile nature of ICN architectures, we expect such a definition to be challenging; static-content popularity approaches usually result in out of date calculations and unutilized cache capacity [14]. Therefore, applying a dynamic-content popularity approach on an ICN architecture, instead of a static one, should be preferred.

Dynamic-content popularity is defined as the number of requests for a content, during a time interval Δ_t [16], [21]. Consequently, the popularity of a content is concluded by comparing its request rates against each other's. A common technique to provide an up to date content popularity pattern is to sort the contents in a decreasing order [5], [15]. A disadvantage, though, deriving from this technique is the constant comparison of the request rates. Towards this direction, we propose a dynamic-content popularity approach that reduces the number of comparisons to a minimum [2].

We have stated the reasons why we chose popularity to be one of the factors that construct our algorithm. We now attempt to explain the reasons why we chose distance to be the combined factor. Similarly to any caching technique, on-path caching is requested to answer the question of where to cache a content. We slightly change the question into whether a content should be cached on a node based on a network metric. Latency reduction is one of the most preferable metrics with regard to network performance, with the number of hops defined as a sufficient estimation of its value [15].

In order to explain the caching algorithm further, we define some notations. Let i denote the node performing the caching decision and j denote the content on which the caching decision is applied. Let $r_{i,j}$ denote the number of requests on node i for content j, with $\sum_{j=1}^{J} r_{i,j}$ being the total number of requests and $d(i, i')$ the distance between nodes i and i', using the number of hops as a metric. We then define the $Prob-PD_{i,j}$ algorithm as follows:

$$Prob-PD_{i,j} = \underbrace{\frac{\frac{r_{i,j}}{\Delta_t}}{\sum_{\forall j \epsilon J} \frac{r_{i,j}}{\Delta_t}}}_{P} \times \underbrace{\frac{d_{n,src}}{d_{dst,src}}}_{D} \tag{1}$$

where, *P* is the dynamic-popularity calculation of content *j*, constructed by the number of requests for content *j* during the time interval Δ_t, divided by the total number of requests during the same time interval, on node *i*. In order to avoid introducing any additional overhead on the operation of a node, we define Δ_t to be the time between the arrival of the first request for content *j* and the satisfaction of it. This way, a content is limited to one comparison only against the rest of the contents, minimizing the complexity that dynamic-popularity calculations apply. The *D* factor, is constructed based on the distance between node *i* and the source serving the request *src*, normalized by the distance between *src* and the node requesting the content *dst*. The *D* factor, represents the benefit of caching a content on node *i*, against the cost of retrieving the content from the original source. Our goal is to examine how beneficial may be the combination of these two factors regarding the ICN on-path caching problem.

IV. Evaluation

In order to provide some initial indications of the performance of the proposed algorithm, we include the results of some early simulations based on the *ndnSIM* simulator [1], an ns-3 module that implements the Named Data Networking (NDN) communication model [12]. The experimental topology is a 5-level binary-tree, with the root being the source of *1000* contents. Any node in the topology, except the root, is able to request a content and cache it in its content store *(CS)*. The capacity of a CS is equal to the size of 10 contents. Contents in a CS are replaced using a LRU policy [3], [12], [18]. Content requests are generated on each node in parallel, following a Zipf distribution of $\alpha\epsilon\{1.0, 1.5\}$. Finally, a shortest path routing protocol is assumed. The results are concluded after the completion of about $3x10^7$ content requests in total.

Using the simulation model described, we compare the performance of the CE^2, *DC*, $FIX(0.90)$, $ProbCache$ and $Prob-PD$ algorithms with regard to the overall cache hit rates, overall cache replacement rates and average content delivery times. Due to space restrictions, we modify the names of two of the approaches into a shorter abbreviation, i.e. *PC* and *PD*, for the ProbCache and Prob-PD algorithms, respectively. The results are presented on Fig.1, for *a=1.0* and on Fig.2, for *a=1.5*. On each figure, the x-axis corresponds to the nodes of the binary-tree topology while the y-axis corresponds to the evaluation metric recorded, e.g. the cache hit rates. One general observation that can be derived from the aforementioned figures is a certain pattern of anomalies on the behavior of the algorithms, e.g. the lower content delivery times, with regard to the simulation topology; one can easily distinguish that the points where these anomalies appear are the same points where the level of the binary-tree updates. Further investigation on this behavior is out of progress.

According to Fig.1, the PD algorithm seems to outperform the rest of the approaches in terms of cache hit rates and cache replacement rates. More precisely, the DC algorithm is scaling in similar rates after the 30^{th} node, with regard to the cache hit rates while the PC algorithm is scaling slightly worse, with regard to the cache replacement rates. As expected, the CE^2 and $P(0.90)$ algorithms result in the lowest cache hit rates and the highest cache replacement rates but they still hold the advantage of the lowest delivery times. However, the same, roughly, delivery times are reached by the PD algorithm.

By comparing Fig.1 and Fig.2, one can conclude that the general trend of the algorithms is preserved for the majority of them, as a increases from 1.0 to 1.5. The most noticeable difference is the behavior of the PC algorithm, resulting in higher cache hit rates and higher content delivery times. The distinctive behavior of the DC algorithm after the 30^{th} node is related to the nature of the algorithm; a content is cached, only at the nodes having the highest number of neighbors along the delivery path. Therefore, nodes 1^{st}-30^{th}, having a DC value equal to three, would be preferred over nodes 31^{st}-62^{nd}, having a DC value equal to one. As such, nodes 31^{st}-62^{nd} would have a lower chance of caching a content originated from the source. However, this also means that the nodes would have more free space, so as to cache contents being found at the 3^{nd} level of the binary-tree hierarchy, resulting in higher cache hit rates and lower replacement rates.

A final point that we owe to highlight is the dependence of the PD algorithm to the nature of the workload, such as the number of contents. This drawback derives from the way that the popularity factor P, is calculated. Considering an example where the number of requests for content j during the time interval Δ_t is 10 and the total amount of requests during the same time interval is 1000, then $P = 0.01$, which would probably result in a non-caching decision. We do recognize that this validation may be important and a different approach to the way popularity is calculated should be considered.

To this end, an alternative option is about to be tested, where P is equal to the number of requests for content j during the time interval Δ_t, divided by the maximum number of requests for any content, during the same time interval. The advantage of this approach lies to the fact that contents are now competing with each other and not with the whole number of requests. Following the same example as before and assuming that the highest number of requests for any content observed is equal to 50, then $P = 0.2$. That way a better approximation of the content's popularity may be provided.

V. Conclusion

In this paper, we have described the existing on-path caching algorithms for the ICN architectural model and categorized them against their properties. We have further proposed a probabilistic caching algorithm, *Prob-PD*, to enhance performance. In order to have some initial results, we evaluated our approach against the CE^2, *DC*, $FIX(0.90)$ and *ProbCache* alternatives. The results indicate, that *Prob-PD* may provide significant gains if certain conditions are met. However, since our model is an early evaluation, further research is necessary to conclude to the suitability of the algorithms and their performance.

References

[1] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnsim: Ndn simulator for ns-3," *Technical Report NDN-0005, Named-Data Networking Project*, October 2012.

[2] F. M. Al-Turjman, A. E. Al-Fagih, and H. S. Hassanein, "A value-based cache replacement approach for information-centric networks," in *Proceedings of the 13th IEEE International Workshop on Wireless Local Networks (WLN '13), Sydney, Australia, October 2013*, pp. 874–881.

[3] S. Arianfar, P. Nikander, and J. Ott, "On content-centric router design and implications," in *Proceedings of the Re-Architecting the Internet Workshop of the ACM CoNEXT Conference*, no. 5, 2010.

[4] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica, and M. Walfish, "A layered naming architecture for the internet," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communications (SIGCOMM '04), Portland, OR, USA, August 2004*, pp. 343–352.

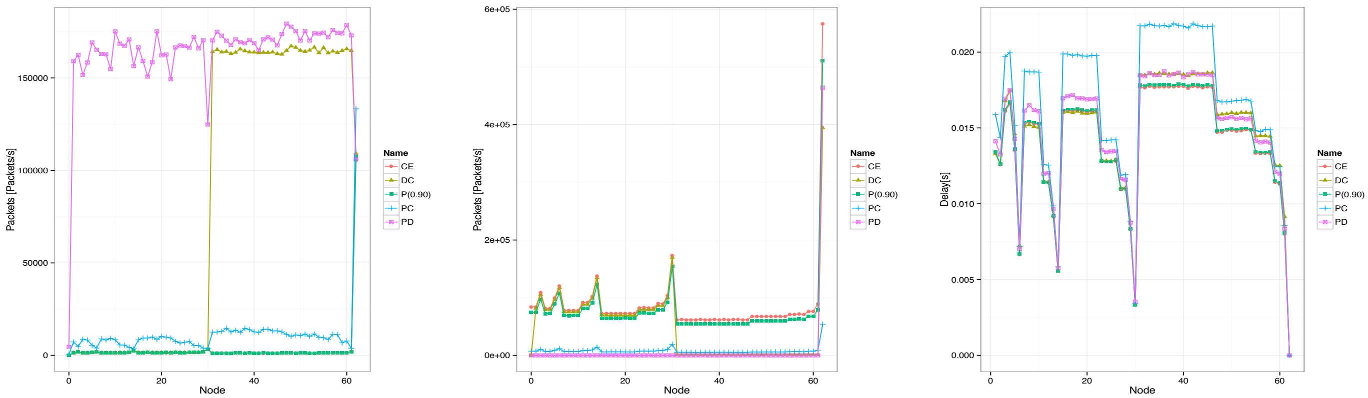

Fig. 1: Metrics for $\alpha = 1.0$, $(a.)$ Cache hit rates, $(b.)$ Cache replacements rates, $(c.)$ Content delivery times

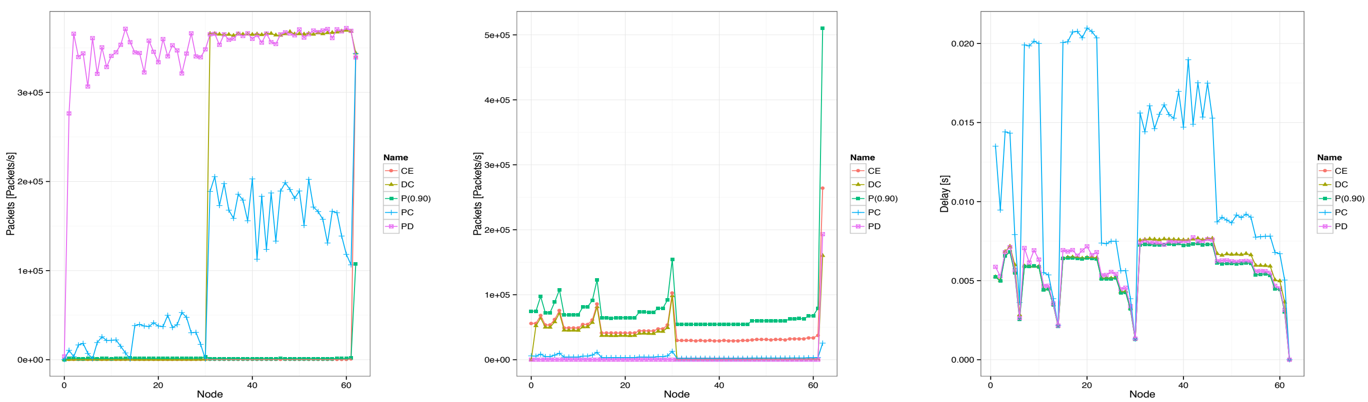

Fig. 2: Metrics for $\alpha = 1.5$, $(a.)$ Cache hit rates, $(b.)$ Cache replacements rates, $(c.)$ Content delivery times

[5] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proceedings of the 29th IEEE Conference on Computer Communication (INFOCOM '10), San Diego, CA, USA, March 2010*, pp. 1–9.

[6] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Modeling data transfer in content-centric networking," in *Proceedings of the 23rd International Teletraffic Congress (ITC'11), San Francisco, CA, USA, September*, 2011, pp. 111–118.

[7] W. Chai, D. He, I. Psaras, and G. Pavlou, "Cache "less for more" in information-centric networks," in *Proceedings of the 11th International IFIP TC 6 Conference on Networking, Brooklyn, NY, USA, May 2012*, pp. 27–40.

[8] H. Che, Y. Tung, and Z. Wang, "Hierarchical web caching systems: Modeling, design and experimental results," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 7, pp. 1305–1314, 2002.

[9] M. Draxler and H. Karl, "Efficiency of on-path and off-path caching strategies in information centric networks," in *Proceedings of the IEEE International Conference on Green Computing and Communications (GreenCom '12), Besancon, France, Novemver 2012*, pp. 581–587.

[10] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox, "Information-centric networking: seeing the forest for the trees," in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks (HotNets-X '11), Cambridge, MA, USA, November 2011*, p. 1.

[11] A. Ioannou and S. Weber, "Towards on-path caching alternatives in information-centric networks," *Technical Report TCD-CS-2014-02*, June 2014.

[12] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT '09), Rome, Italy, December 2009*, pp. 1–12.

[13] S. Jamin, C. Jin, A. Kurc, D. Raz, and Y. Shavitt, "Constrained mirror placement on the internet," in *Proceedings of the 20th IEEE Conference on Computer Communication (INFOCOM '01), Anchorage, AK, USA, April 2001*, pp. 31–40.

[14] T. Janaszka, D. Bursztynowski, and M. Dzida, "On popularity-based load balancing in content networks," in *Proceedings of the 24th International Teletraffic Congress (ITC'12), Krakow, Poland, September 2012*, pp. 1–8.

[15] J. Kangasharju, J. Roberts, and K. Ross, "Object replication strategies in content distribution networks," *Elsevier Journal on Computer Communications*, vol. 25, no. 4, pp. 376–383, 2002.

[16] N. Laoutaris, O. Telelis, V. Zissimopoulos, and I. Stavrakakis, "Distributed selfish replication," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 12, pp. 1401–1413, 2006.

[17] I. Psaras, W. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *Proceedings of the 2nd Workshop on Information-Centric Networking, Orlando, FL, USA, March 2012*, pp. 55–60.

[18] D. Rossi and G. Rossini, "Caching performance of content centric networks under multi-path routing (and more)," Telecom ParisTech Ecole, Tech. Rep., November 2011.

[19] G. Rossini and D. Rossi, "On sizing ccn content stores by exploiting topological information," in *Proceedings of the 1st Workshop on Emerging Design Choices in Name-Oriented Networking (NOMEN '12), Orlando, FL, USA, March 2012*, pp. 280–285.

[20] V. Sourlas, G. Paschos, P. Flegkas, and L. Tassiulas, "Caching in content-based publish/subscribe systems," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '09), Honolulu, HI, USA, November 2009*, pp. 1–6.

[21] V. Sourlas, P. Flegkas, L. Gkatzikis, and L. Tassiulas, "Autonomic cache management in information-centric networks," in *Proceedings of the IEEE Network Operations and Management Symposium (NOMS '12), Maui, HI, USA, April 2012*, pp. 121–129.

[22] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys and Tutorials*, no. 1–26, 2013.