Hash-routing Schemes for Information Centric Networking

Lorenzo Saino, Ioannis Psaras and George Pavlou
Department of Electrical and Electronics Engineering
University College London
London, UK
{l.saino, i.psaras, g.pavlou}@ucl.ac.uk

ABSTRACT

Hash-routing has been proposed in the past as a mapping mechanism between object requests and cache clusters within enterprise networks.

In this paper, we revisit hash-routing techniques and apply them to Information-Centric Networking (ICN) environments, where network routers have cache space readily available. In particular, we investigate whether hash-routing is a viable and efficient caching approach when applied outside enterprise networks, but within the boundaries of a domain.

We design five different hash-routing schemes which efficiently exploit in-network caches without requiring network routers to maintain per-content state information.

We evaluate the proposed hash-routing schemes using extensive simulations over real Internet domain topologies and compare them against various on-path caching mechanisms. We show that such schemes can increase cache hits by up to 31% in comparison to on-path caching, with minimal impact on the traffic dynamics of intra-domain links.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Distributed networks*

General Terms

Design, Performance, Experimentation

Keywords

ICN; off-path caching; cache-aware routing; hash-routing

1. INTRODUCTION

In-network caching is expected to improve the performance of content transfers in Information-Centric Networks (ICN), by allowing any network element to become a temporary content server. As a result, requests can be fulfilled by any element along the path if the content is cached locally. This approach, however, bears the challenge of how to manage and co-ordinate such distributed storage infrastructure in a scalable, efficient and cost-effective way.

According to [9], there are three main challenges with regard to the setup, co-ordination and management of a ubiquitous in-network caching system. Firstly, there is a *cache placement* challenge, which addresses the issue of which nodes within a domain are upgraded to in-network caches. This constitutes mainly a network planning problem, as the decision on which nodes to upgrade should take into account the domain's network topology, traffic characteristics and position within the Internet hierarchy [8], [13]. In this study, we assume that a domain has already deployed caches internally and focus on the two remaining challenges: the *content placement* challenge, which addresses how to distribute contents across in-network caches of a domain, and the *request-to-cache routing* challenge, which determines how content requests are resolved to the corresponding cache nodes.

We tackle these challenges by revisiting *hash-routing* techniques, which in the past have been used in enterprise networks ([15], [20], [18]) to determine content placement and retrieval in order to minimize latency. In hash-routing, content placement and request-to-cache routing are governed by a hash function that maps content identifiers to cache locations. In the context of Information Centric Networks we adopt similar techniques, but we focus on optimizing the utilization of available in-network caching space and therefore maximize the likelihood of a cache hit.

We target domain-wide ICN deployments and assume that some (hierarchical or flat) content naming scheme is in place. Our design also requires that edge-domain routers implement a hash function that determines both the content placement and the request-to-cache routing process. When the edge routers of a domain receive a content request, they calculate the hash of the content identifier and redirect it to the responsible cache. In the case of a cache hit, the content is returned to the client, otherwise, the request is forwarded towards the original server. Similarly, incoming contents are forwarded according to the hash of their identifier. The main concept underpinning our approach is that *if a content exists within one of the domain's caches, then it will be in the cache calculated by the hash function.*

Our contribution in this paper is twofold. First, we revisit *hash-routing* techniques in the context of ICN environments, where caches are placed within an ISP network rather than in clusters within enterprise networks. In doing so, we investigate the impact of re-routing request and content packets within a domain in terms of fluctuations in link load.

ICN'13, August 12, 2013, Hong Kong, China.

ACM 978-1-4503-2179-2/13/08 ...$15.00.

Building on traditional hash-routing mechanisms, we then propose two novel hybrid approaches which achieve optimal trade-offs between cache hit rate and intra-domain link load.

We evaluate the proposed hash-routing algorithms through simulation using real intra-domain topologies and demonstrate that hash-routing can reduce inter-domain traffic by up to 31% as a result of increased cache hits. This reduction comes at the cost of a small increase in network load across intra-domain links for traditional hash-routing techniques. Differently, the proposed hybrid schemes achieve a similar reduction of inter-domain traffic without a significant increase in intra-domain traffic.

The rest of the paper is organized as follows. In Section 2, we motivate our study and discuss related work. In Section 3, we formulate the design space under which hash-routing is applied and introduce two hybrid approaches. In Section 4, we evaluate the performance of hash-routing in comparison to on-path caching. Finally, in section 5, we summarize our findings and draw conclusions.

2. MOTIVATION AND BACKGROUND

Content placement and request-to-cache routing strategies for distributed caching systems can broadly be ascribed to two distinct categories:

- **On-path content placement with opportunistic request-to-cache routing.** According to this approach, contents are cached as they travel through the network by any on-path cache [6] or a subset of traversed caches [14], [2]. Content requests are forwarded towards the content source according to the underlying forwarding rules. Contents are consequently retrieved from caches in an opportunistic manner. On-path caching has attracted wide attention in the ICN research community and spawned interest in topics such as reducing caching redundancy [14], [22], caching prioritization by popularity assessment [3], [12] and content locality [19]. On-path caching has also been investigated in the past, in the context of overlay (hierarchical) web-caching systems [10]. However, research in that domain focused on issues surrounding cache placement rather than content placement [8], [21].
- **Off-path content placement with co-ordinated request-to-cache routing.** In contrast to the previous approach, off-path content placement operates according to predefined rules that assign contents to caches (e.g., [1]). In turn, request-to-cache routing must also adhere to the same rules in order to retrieve contents from caches in a co-ordinated manner [17], [18], [15] [7]. *Hash routing* has been one of the prominent solutions proposed in this domain [15].

Both on-path and off-path caching present trade-offs. On-path content caching requires less co-ordination and management, but may provide limited gains. Conversely, off-path content placement and retrieval can attain higher hit rates at the cost of extra co-ordination and communication overhead. Co-ordination overhead refers to the decision making process of where to cache incoming contents, as well as to the forwarding rules that (re-)direct incoming requests to cached contents [17], [23]. Communication overhead refers to the redirection of requests and the retrieval of contents from off-path caches, which involve possibly traversing longer paths and therefore, increasing network load [1], [17], [22], [23].

Closer to our work in the ICN area there exist recent proposals to co-ordinate content placement and request routing in the new, ubiquitous caching system [22], [11], [5], [4]. In [5], a proposal that applies to CCN/NDN [6], the authors propose a content placement and retrieval technique based on popularity assessment in order to avoid overloading specific links. The location of the objects is kept in an *Availability Information Base* (AIB) table. The study focuses on the scattering efficiency of the algorithm based on its popularity assessment mechanism. We argue that a hash function inherently supports scattering of popular contents without incurring any further communication overhead.

SCAN [11] is a hybrid on-path/off-path content retrieval technique, where only nodes a few hops away from the shortest path are considered as potential sources of content (i.e., caches) upon an incoming content request. Although this mitigates concerns regarding extra delays due to long off-path routes, it limits potential cache hits to only the immediacy of the shortest path. In contrast, our proposal takes advantage of all the available caches within a domain. Finally, [22] and [4] use co-ordination techniques between the data and the control plane to place contents and re-direct requests to caches. We argue that such approaches introduce considerable amounts of overhead and delay and as such result in an inherently less scalable solution.

3. ICN APPROACHES TO HASH-ROUTING

The hash-routing schemes proposed in this paper require edge-domain routers and cache nodes to implement a hash function which maps content identifiers to cache nodes. This function is used by cache nodes to know what contents they can store and by edge routers to route requests and contents to the relevant cache node (see fig. 1). As a result of this approach, each content object can be cached in a domain at most once, thus preventing redundant replication of cached contents and resulting in a more efficient utilization of cache space. This approach also allows edge routers to forward content requests to the designated cache directly, without performing any lookup. In addition, this is performed without requiring any sort of inter-cache co-ordination since the hash function is computed in a distributed manner by edge routers and caches, thus achieving great scalability.

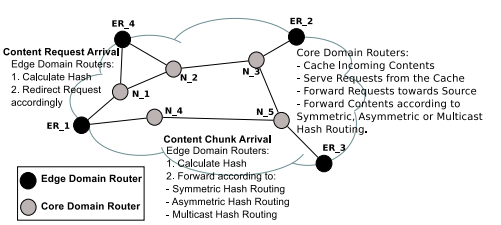


Figure 1: Hash-routing Functional Architecture

The hash function maps a content object identifier (flat or hierarchical) to a value $x_i : i \in [1, N]$ where N is the number of cache nodes in the domain. Such function does not need to produce a cryptographic hash. In fact, it is desirable for its output to be produced with minimal processing. For example a modulo hashing of the content identifier would be a suitable candidate.

Ideally, the hash function should be capable of mapping contents to cache nodes so that the load of caches is evenly spread. In our experimentations we verified that this is achieved if the content popularity is Zipf-distributed with skewness parameter $\alpha < 0.95$, which is pretty much the design space of our routing schemes. However, as we show in section 4, hash-routing schemes perform well even for greater values of α.

In the rest of this section, we provide detailed description of three different instantiations of hash-routing schemes for in-network caching environments, namely *symmetric*, *asymmetric* and *multicast* hash-routing. In addition, based on our observations from these three schemes, we propose two *hybrid approaches* which combine the symmetric and multicast and the asymmetric and multicast approaches, respectively. The hybrid approaches take advantage of the strengths of each of their respective parts.

3.1 Symmetric, Asymmetric and Multicast Hash Routing Schemes

To explain how symmetric, asymmetric and multicast hash-routing operate, we use the example depicted in fig. 2.

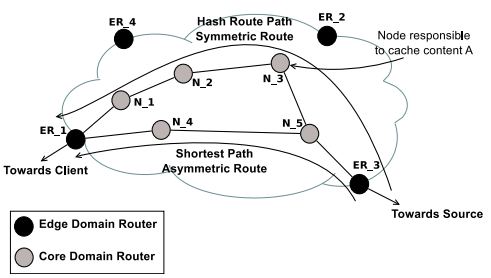


Figure 2: Symmetric and Asymmetric Hash-routing

Consider a request for content A that arrives in the domain from edge router ER_1. In order to fetch the requested content from the origin server, ER_1 would normally forward the request to node ER_3 via N_4 and N_5. However, the hash function, calculated at ER_1, points to node N_3, which is responsible for caching the requested content. Therefore, the request is sent through the path $N_1 - N_3$. If the content is in the cache of node N_3, it is delivered to the requesting client through nodes $N_1 - N_3$ to ER_1. Otherwise, this node forwards the request towards the original source (i.e., to node ER_3 via N_5).

When content A comes back from the origin server, it enters the domain from ER_3. At this point, edge router ER_3 has three options:

- **HR Symm:** ER_3 forwards content A through the path traversing the cache responsible to hold this content (i.e., node N_3); the content is replicated and cached at node N_3 and it is also sent back to the requesting client through path $N_2 - N_1 - ER_1$. We call this option *HR Symm*, as the forward and the return paths are *symmetric*.
- **HR Asymm:** ER_3 sends content A back to the requesting client through the shortest path (i.e., path $N_5 - N_4 - ER_1$). In this specific case, content A is not cached within this domain, since the designated cache (N_3) is not on the path. We call this option *HR Asymm*, as the forward and the return paths are *asymmetric*.
- **HR Multicast:** ER_3 replicates content A and sends one copy to node N_3 to cache it and one copy back towards the client through the shortest path (i.e., path $N_5 - N_4 - ER_1$). We call this option *HR Multicast*, as the content is effectively multicast from the edge router towards both the responsible caching node and the client.

It is worth noting that if the cache responsible for a given content is on the path from content ingress node and receiver, then symmetric, asymmetric and multicast approaches exhibit exactly the same behaviour.

Symmetric and multicast hash-routing schemes have the potential of providing the best performance in terms of cache hits even in presence of low traffic because every time a content object enters the domain it is delivered to the responsible cache. However, this comes at a cost of potentially increasing the load of intradomain network links as a consequence of the detouring of content packets to be delivered to the cache. While symmetric and multicast achieve identical performance in terms of cache hit, their performance may differ substantially in terms of delay and link load depending on network topology.

The asymmetric hash-routing scheme mitigates this problem by deflecting the route of only request packets which have a considerably smaller size than content packets, while content packets are always routed via the shortest path from source to destination and cached only if the responsible cache is on the path. Nevertheless, this option may reduce cache hits, as later requests for the same content will have to be retrieved from the origin server. This would inevitably result in traversing potentially *expensive* inter-domain links whenever this content is requested.

3.2 Hybrid Hash-routing Schemes

As shown in the previous section, one of the main design issues of off-(shortest)-path caching algorithms is the extra number of hops that request and content packets have to travel before reaching their destination.

In order to address this problem, we present here two hybrid hash-routing schemes, which we called *Asymmetric-Multicast* and *Symmetric-Multicast*.

These schemes have been designed with the objective of reducing the path stretch introduced by content packet detouring by dynamically selecting the most appropriate content forwarding strategy based on the location of source, cache and receiver nodes.

3.2.1 Hybrid Symmetric-Multicast Hash-routing

Both symmetric and multicast hash-routing schemes can achieve high cache hits but at a cost of potentially increasing intradomain link load. However, the increase in link load caused by each of the two schemes depends on the network location of the various nodes involved. For example, within the scenario depicted in fig. 1, if content object entering the domain via ER_3 was directed to client ER_2 and the designated cache was at N_4, then multicast delivery would be more efficient. Differently, if the designated cache was N_3, symmetric delivery would be preferable.

To address these limitations, we propose a hybrid symmetric-multicast hash-routing approach (*HR Hybrid SM*) in which

the edge router from which the content packet enters the domain (ER_3 in fig. 1) decides whether to deliver the packet using the symmetric or multicast scheme depending on the location of cache and destination nodes so that the path stretch is minimized.

More specifically, this scheme operates as follows. The request packet signals to the edge router (ER_3 in our case) the number of hops from the incoming edge node (ER_1) to the node responsible for caching this content (N_3), as well as the distance (in terms of hops) from N_3 to ER_3. Node ER_3 makes the decision of whether to follow symmetric or multicast hash-routing based on the value of $\Delta_{SM} = (d_{(ER_1-N_3)} + d_{(N_3-ER_3)}) - d_{(ER_3-ER_1)}$. If Δ_{SM} is positive, then multicast hash-routing is chosen. Otherwise, symmetric hash-routing is used.

3.2.2 Hybrid Asymmetric-Multicast Hash-routing

As intuitively argued above, asymmetric hash-routing, by following the shortest return path to the client, achieves low delivery delay and link load. However, it may end up caching a limited number of content objects within the network's caches, which can may lead to lower cache hit rates, thus attenuating the effect of in-network caching. On the other hand, multicast hash-routing achieves both low delivery delay (as one copy of the content follows the shortest path to the client), and high cache hit rates (as it replicates contents and sends one copy to the responsible cache), but it may also increase the load of network links.

To address these limitations, we propose a simple combination of the two approaches (*HR Hybrid AM*) providing a better tradeoff among cache hits, delay and link load. Our proposal is based on the *path stretch* between the edge domain router that receives the content on the return path and the node/cache that is responsible for caching this content according to the hash function.

This scheme operates, with reference to fig. 1, as follows. When node ER_3 receives content A: *i)* sends a copy of the content towards the requesting client through the shortest path (asymmetric route) and *ii)* compares the path stretch to the node responsible for caching this content (in our case node N_3[1]) to the *diameter* of the network D. If the length of the path to N_3 is smaller than a predefined fraction k of the diameter D of the network, then ER_3 replicates the content and sends one copy to the responsible cache too. Otherwise, the content is not cached at this time.

For the purposes of this study, the fraction of the diameter of the network for which contents are replicated has been set to $k = 0.2$. We note however, that we have explored different settings and we report that this is a setting related to the characteristics of the topology and therefore, has to be investigated in relation to this.

4. PERFORMANCE EVALUATION

4.1 Methodology and Setup

We evaluated the performance of the off-path hash-routing schemes presented above through flow-level simulations using *Icarus*[2], a simulator based on the *Fast Network Simulation Setup* (FNSS) toolchain [16]. Simulations have been carried out on four real topologies: GEANT (European academic network), GARR (Italian academic network), WIDE (Japanese academic network) and Tiscali (pan-European commercial ISP). Network routers have been assigned constant cache sizes. Content requests have been modelled as Poisson processes while content popularity has been modelled with a Zipf distribution.

We analyzed the performance of the hash-routing schemes by focusing on two specific metrics: the *cache hit ratio* and the *average load* on inter- and intra-domain links. The cache hit ratio corresponds to the fraction of content requests served by caches deployed within the ISP network. This metric reflects the capability of the caching scheme to reduce the amount of redundant inter-domain traffic. On the other hand, the average link load has been measured with the objective of understanding the impact of the route deflection introduced by off-path caching schemes on the intra-domain link utilization.

These two metrics have then been measured for all four topologies under varying conditions of cache sizes and content popularity distribution skewness (represented by the Zipf α parameter). More specifically, results have been evaluated for four values of *cache to content population ratio* C (i.e. cumulative size of network caches as fraction of the total content population), ranging from 0.04% to 5% and for values of α ranging from 0.6 to 1.1. We note that we do not expect hash-routing to be applied to the entire amount of traffic that a domain serves. Instead, we envision that service level agreements between content providers and ISPs will result in favourable treatment of a fraction of traffic.

In order to get a better understanding of the behaviour of the hash-routing schemes within the wider spectrum of in-network caching strategies, we compare their performance also with on-path caching techniques. As benchmarks, we use ubiquitous caching and *ProbCache* [14]. All caching schemes (both on- and off-path) have been evaluated assuming that contents are evicted according to a Least Recently Used (LRU) policy.

4.2 Evaluation results

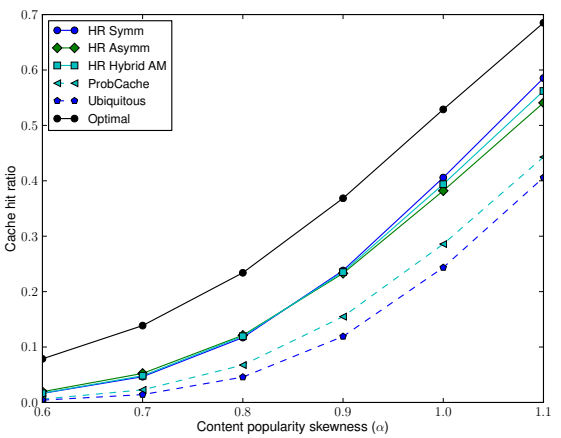
The results of our evaluation are depicted in fig. 3. Fig. 3a shows the sensitivity of the cache hit ratio against variations of α in the GEANT topology with $C = 0.2\%$. As it can be seen from this graph and as somewhat expected, all off-path caching schemes outperform the on-path ones. In particular, among off-path caching schemes, symmetric hash-routing achieves the greatest hit ratio for all values of α considered. *HR Multicast* and *HR Hybrid SM* are not visible in this figure as they perform identically to the symmetric hash-routing scheme and therefore, their data are superimposed by the *HR Symm* line plot.

It should be noted that, although all caching schemes provide similar performance in terms of cache hit ratio, their performance in terms of link load is widely different. In fact, as shown in fig. 3c, while *HR Asymm* and *HR Hybrid AM* impact only marginally the link load in comparison to a no-cache scenario, all other off-path schemes have a much higher impact.

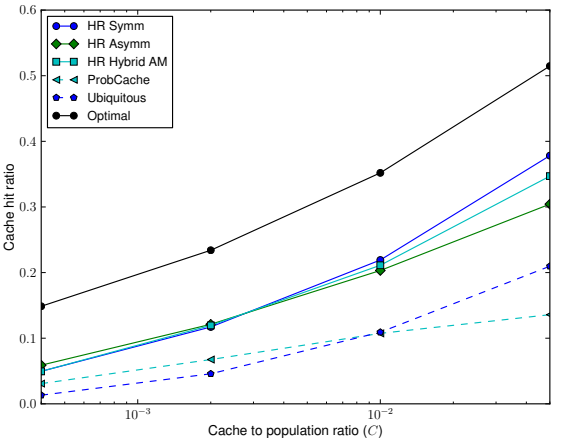
The situation is similar when we evaluate the performance against varying values of *cache to population ratio* while maintaining a constant value of the Zipf α parameter (equal to 0.8) (see fig. 3b and 3d). Off-path hash-routing still clearly outperforms on-path caching strategies. However,

[1]The request message signals the distance to the node responsible to cache the incoming content (N_3) back to the edge node (ER_3).

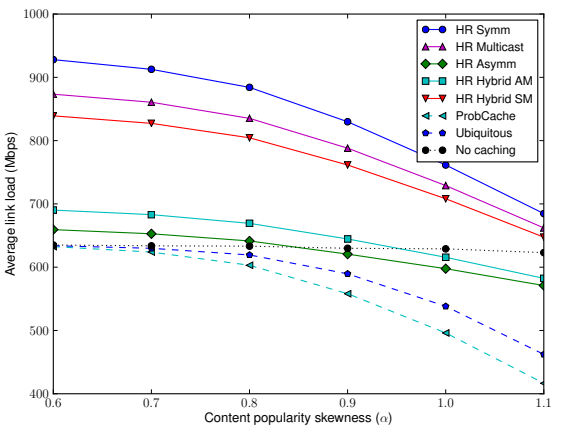
[2]`http://www.ee.ucl.ac.uk/~lsaino/software/icarus`



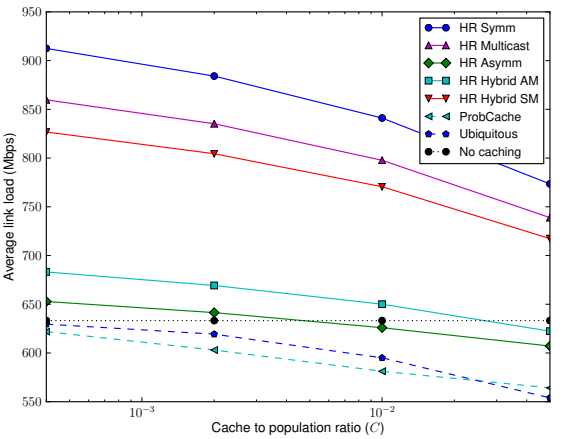
(a) Cache hit ratio vs α - GEANT, $C = 0.2\%$



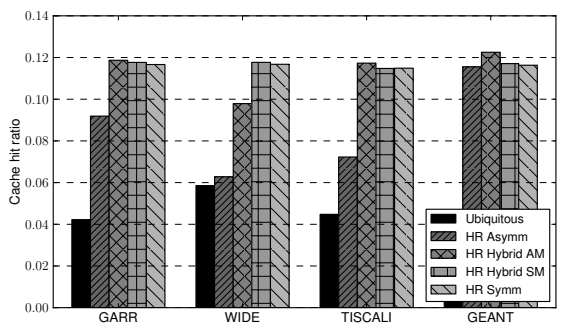
(b) Cache hit ratio vs C - GEANT, $\alpha = 0.8$



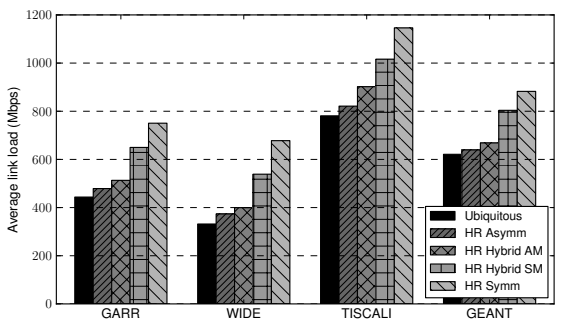
(c) Link load vs α - GEANT, $C = 0.2\%$



(d) Link load vs C - GEANT, $\alpha = 0.8$



(e) Cache hit ratio vs topology - $C = 0.2\%$, $\alpha = 0.8$



(f) Link load vs topology $C = 0.2\%$, $\alpha = 0.8$

Figure 3: Performance of off-path and on-path caching schemes in terms of cache hit ratio and intradomain link load for various topologies, content popularity skewness (α parameter) and cache sizes (C parameter)

as the *cache to population ratio* increases (i.e., more caching space is available within the network), the difference between on-path and off-path caching declines. This is also the case with the performance difference between ubiquitous caching and *ProbCache* [14]. In fact, as we also showed in [14], as the available cache space increases, the need for probabilistic multiplexing of contents along the path decreases.

We report, although do not present here due to space limitations, that in some topologies, for values of α greater than 1.2 and large caches, on-path caching (and in particular *ProbCache*) outperforms the *HR Asymm* scheme in terms of cache hit ratio. This is caused by the limited amount of cachable traffic traversing some cache nodes in *HR Asymm*, whose effects become more severe as the cache size and content distribution skewness increase.

Another very important factor that greatly impacts the performance of both on-path and off-path caching schemes is the network topology. In figs. 3e and 3f, we present the

cache hit ratios and the intradomain network load for the four topologies listed above.

There are three main observations from these results. First, the *HR Asymm* approach is occasionally inconsistent in terms of cache hits. This was expected, given the randomness according to which contents are cached within the domain's caches in this case. Second, the rest of the hash-route approaches (i.e., both the symmetric and the hybrid ones) are pretty stable in terms of successful re-directions (i.e., cache hits). Third, in terms of internal network load, all hash-route approaches seem to have pretty stable trends. However, *HR Symm* and *HR Hybrid SM* hash-routing are the most demanding in terms of load imposed to the network's links.

We have monitored the load on each individual link of every topology when applying both off-path hash-routing techniques and on-path probabilistic caching ([14]) and compared it to the case where no caches are deployed in the network. In all cases, we have found that in case of highly skewed popularity (i.e., high Zipf α), some links become very loaded, as they serve the popular contents for all clients within the domain. This is not the case, for example, when no caches are available and clients are receiving contents from several different parts of the network. For moderately high popularity skewness we report the following. For on-path caching, the link load remains at similar levels as in the case of no caching. For off-path hash-route caching, we have found that around 60-70% of links carry the same amount of traffic (difference below 5%). Around 10-30% of links have to carry approximately 10-20% more traffic, while a small percentage of links (less than 10%) might have to carry 40% more traffic than they do under no caching conditions.

However, this extra load on the domain's links has to be considered in conjunction with the fact that if not served by this domain's caches, contents will need to be fetched from domains possibly many AS-hops away. Therefore, the trade-off between increased internal link load within one domain, might be counter-balanced by the gains in terms of overall delivery delay or provider costs.

5. CONCLUSIONS

In this paper, we have revisited past cache-aware routing techniques using hash functions, commonly known as hash-routing and adapted them to the specific requirements of an ICN environment. We have devised three simple approaches to hash-routing: *symmetric*, *asymmetric* and *multicast* and proposed two hybrids of these: *symmetric-multicast* and *asymmetric-multicast*. We have extensively evaluated the performance of these off-path caching techniques and found that they are promising candidates for ICN domains. In fact, the difference in terms of cache hits, and therefore, in terms of (potentially expensive) inter-domain traffic is considerable and reaches up to 31% of the overall traffic. However, the behaviour of some of the hash-routing techniques (e.g., asymmetric, as well as the hybrid of symmetric-multicast) depends strongly on the topology on which it is applied. We plan to extend our work to consider multiple collaborating peering domains (e.g., [13]) and further evaluate the performance of both on-path and off-path cache-aware routing techniques in case of realistic Internet traces. This will help model the trade-off between increased internal load and reduced latency.

6. ACKNOWLEDGMENTS

The research leading to these results was funded by the EU-Japan initiative under EC FP7 Grant Agreement no. 608518 and NICT Contract no. 167 (the GreenICN project).

7. REFERENCES

[1] S. Bhattacharjee, K. L. Calvert, and E. W. Zegura. Self-organizing wide-area network caches. In *IEEE INFOCOM*, 1998.

[2] W. Chai, D. He, I. Psaras, and G. Pavlou. Cache 'less for more' in information-centric networks. In *IFIP NETWORKING*. 2012.

[3] K. Cho and et al. Wave: Popularity-based and collaborative in-network caching for content-oriented networks. In *INFOCOM NOMEN Workshop*, 2012.

[4] S. Eum, K. Nakauchi, M. Murata, Y. Shoji, and N. Nishinaga. Catt: potential based routing with content caching for icn. In *ACM SIGCOMM ICN Workshop*, 2012.

[5] S. Guo, H. Xie, and G. Shi. Collaborative forwarding and caching in content centric networks. In *In Proceedings of IFIP Networking*, 2012.

[6] V. Jacobson and et al. Networking named content. In *Proceedings of CoNEXT*, New York, NY, USA, 2009. ACM.

[7] K. Katsaros, G. Xylomenos, and G. C. Polyzos. Multicache: An overlay architecture for information-centric networking. *Comput. Netw.*, 55(4):936–947, Mar. 2011.

[8] P. Krishnan, D. Raz, and Y. Shavitt. The cache location problem. *IEEE/ACM Trans. Netw.*, 8(5), 2000.

[9] D. Kutscher and et al. Icn research challenges. *IRTF, draft-kutscher-icnrg-challenges-00*, Februady 2013.

[10] N. Laoutaris, H. Che, and I. Stavrakakis. The lcd interconnection of lru caches and its analysis. *Perform. Eval.*, 63(7), July 2006.

[11] M. Lee, K. Cho, K. Park, T. T. Kwon, and Y. Choi. Scan: Scalable content routing for content-aware networking. In *IEEE ICC 2012*, 2012.

[12] J. Li and et al. Popularity-driven coordinated caching in named data networking. In *Proceedings of ANCS*, New York, NY, USA, 2012. ACM.

[13] V. Pacifici and G. Dan. Content-peering dynamics of autonomous caches in a content-centric network. In *IEEE INFOCOM*, 2013.

[14] I. Psaras, W. K. Chai, and G. Pavlou. Probabilistic in-network caching for information-centric networks. In *Proceedings ICN Sigcomm workshop*, pages 55–60, New York, NY, USA, 2012. ACM.

[15] K. W. Ross. Hash routing for collections of shared web caches. *Netwrk. Mag. of Global Internetwkg.*, 11(6):37–44, Nov. 1997.

[16] L. Saino, C. Cocora, and G. Pavlou. A toolchain for simplifying network simulation setup. In *Proceedings of SIMUTOOLS*, 2013.

[17] R. Tewari, M. Dahlin, H. M. Vin, and J. S. Kay. Design considerations for distributed caching on the internet. In *Proceedings of ICDCS*, Washington, DC, USA, 1999.

[18] D. G. Thaler and C. V. Ravishankar. Using name-based mappings to increase hit rates. *IEEE/ACM Trans. Netw.*, 6(1):1–14, Feb. 1998.

[19] G. Tyson and et al. A trace-driven analysis of caching in content-centric networks. In *Proc of ICCCN*, 2012.

[20] V. Valloppillil and K. W. Ross. Cache array routing protocol v1.0. *Internet Draft draft-vinod-carp-v1-03.txt*, February 1998.

[21] J. Wang. A survey of web caching schemes for the internet. *SIGCOMM Comput. Commun. Rev.*, 29(5), Oct. 1999.

[22] Y. Wang and et. al. Advertising cached contents in the control plane: Necessity and feasibility. In *INFOCOM NOMEN Workshop 2012*, pages 286–291, 2012.

[23] D. Wessels and K. Claffy. Rfc 2186, icp: Internet cache protocol. *IETF*, September 1997.