Minimizing Bandwidth on Peering Links with Deflection in Named Data Networking

Damien Saucez, Anshuman Kalla, Chadi Barakat, Thierry Turletti

Minimizing Bandwidth on Peering Links with Deflection in Named Data Networking

Damien Saucez, Anshuman Kalla, Chadi Barakat, Thierry Turletti
INRIA – Sophia Antipolis – France

ABSTRACT

Data dissemination and service access constitute the primary usage of the Internet today, whereas the associated protocols (TCP/IP) are designed for point-to-point communication. Content-Centric Networking (CCN) has been proposed to overcome this limitation in the existing Internet protocols. CCN omits the notion of host and location by establishing contents as the first class citizens of the network. Furthermore, and to improve content delivery, CCN advocates in-network caching, i.e., to cache contents on the path from content providers to requesters. This on-path caching achieves good overall performance, however, and as we demonstrate in this paper, this strategy is far from being the optimal inside a domain. Instead, we introduce the notion of off-path caching by allowing deflection of the traffic off the optimal path towards off-path caches available across the domain. Off-path caching improves the global hit ratio and permits to reduce the peering links' bandwidth usage without impacting the delay or overloading intra-domain links. We validate our theoretical results with simulations on real topologies and we propose an optimal as well as heuristic off-path caching techniques.

1. INTRODUCTION

Internet applications have drastically changed since its inception and content delivery and retrieval are becoming the most prominent usage. Unfortunately, the Internet with its IP and TCP protocols was designed for point-to-point communication, causing a clear gap between the technology and its current usage therefore reducing the efficiency of communication.

To better fit the technology with the state-of-the-art usage, *Named Data Networking* (NDN) has been proposed [9]. The concept of NDN is to remove the notion of end-to-end communication by accessing contents directly with their names. Different architectures have been proposed like CCN [2], DONA [3], PSIRP [4], or 4WARD [7]. Even though they differ in their implementation and conception, these architectures all rely on assigning a universal name (hierarchical or flat) to each content rather than identifying it by the IP address of the machine hosting it. The name of a content is directly used to fetch it, while with "legacy Internet", the name must be resolved into the IP address of the hosting machine. In this paper, we focus on CCN [2] but our study and findings can be applied to the other NDN proposals as well.

CCN leverages the use of caching to make popular contents available close to their consumers, hence reducing the load on the network and improving QoS. In the studies so far, caching is opportunistic at routers on the path traversed by contents. It has been shown that this solution reduces the overall content retrieval time and bandwidth consumption [6, 1]. Indeed, requests for the same content will only traverse a portion of the path until finding a cached entry. However, this on-path caching inside an Autonomous System (AS) (or a domain) is not optimal in terms of peering links' bandwidth usage, if we consider that the delay to retrieve a content via a peering link is much higher than the delay of retrieving a content inside the AS (wherever it is) and that the cost of using links inside the AS is negligible compared to the cost of using a peering link (e.g., low bandwidth or high monetary cost on the peering links). However, on-path caching results in an uncontrolled duplication of contents within an AS (single content being cached at numerous caches), hence wasting cache space and reducing overall hit rate. Caches within an AS can coordinate their effort so that they avoid replicas and use the freed space to cache more contents, which should lead to a better overall hit rate and lower traffic on peering links. At the same time, this coordination of caches inflate intra-domain paths resulting in a higher internal bandwidth usage. The question is then *how to perform caching within an AS such that the use of peering links is minimized while keeping the AS links' usage below their nominal capacities.*

To minimize the peering links' usage of an AS network operating CCN, we propose to allow traffic deflection (i.e., indirect routing) within the AS in such a way that all requests related to one particular content traverse the same cache (or set of caches). Therefore, if the content is cached, then it is certain that whichever client (associated with the AS) asks for it, the request

will eventually arrive at the specific cache handling the content. This avoids content replication and allows optimal usage of overall cache space, hence maximizing the overall hit rate and minimizing the usage of peering links. To achieve our goal we explore two different techniques. First, we propose an optimal off-path caching technique where (*i*) the N most popular contents are estimated (N being the global cache size of an AS) (*ii*) the optimal placement of these N most popular contents is performed by formulating and solving a linear optimization problem and (*iii*) a workable implementation of this technique is presented. We prove with simulations on real ISP topologies issued from Rocketfuel [8, 5] that optimal off-path caching performs far better than on-path caching. Second, we put forward a random placement heuristic off-path caching technique that utilizes a hash function to deterministically determine the cache at which the content must be placed. We name this technique *Caching All Contents by Hashing* (CACH). CACH is simple to implement and generates less overhead for deflection of traffic toward off-path caches compared to optimal off-path caching. Further, we show with simulations that CACH performs close to optimal off-path caching in terms of hit rate and peering links' usage.

The remainder of this paper provides necessary background on CCN in Sec. 2. Sec. 3 discusses optimal off-path caching with an optimization to minimize the peering link bandwidth usage and its deployment in Sec. 3.2. Heuristic CACH is described in Sec. 4. Finally, the proposition is evaluated with simulations on real topologies in Sec. 5 and Sec. 6 concludes this work.

2. CCN IN A NUTSHELL

Content-Centric Networking proposed by Jacobson et al. in [2] has emerged as a promising future networking paradigm with the key idea of establishing the content, identified solely by its name, as self-sustaining entity while excluding the notions of host and location. Entire communications in CCN are carried out by two types of packets (*i*) *Interest*, being generated and sent by a CCN client when requesting a content, (*ii*) *Data*, containing the requested content, being sent in response to an Interest. Every CCN router maintains three data structures: (*i*) *Pending Interest Table* (PIT) that maintains list of interested faces for each unsatisfied content request, (*ii*) *Forwarding Information Base* (FIB) that holds a list of faces that can potentially serve a requested content and (*iii*) *Content Storage* (CS) that retains a copy of content (in the limit of available storage) even after forwarding it to requesting faces. CS enables in-network caching that transforms every CCN router into a legitimate source of data. An Interest packet is forwarded by intermediate CCN routers based on their FIBs, while updating their PITs so that to be able to later match content to requesting faces. In return, a content is sent by the origin server or any CCN router encountered over the path followed by the Interest packet and holding a copy of the requested content. The Data packet back-tracks the path laid down by the Interest packet at each intermediate router and the carried content can be eventually cached at those routers.

3. OPTIMAL OFF-PATH CACHING

To minimize the peering link bandwidth usage, one needs to enforce that one content is cached at maximum one place inside a CCN domain (i.e., an AS) and we aim at minimizing the traffic over its peering links with other domains by maximizing the hit rate of requests within the domain. We are motivated by the fact that (*i*) resources of a domain are limited and has to be optimally utilized, and (*ii*) the cost of using a peering link is high compared to the cost of operating local links. This global idea is in line with CCN policy which emphasizes on the possibility to cache popular contents inside the AS [2]. In order to minimize peering links' bandwidth usage by maximizing the hit rate, we deflect the traffic within an AS towards the caches holding cached copies of requested data. Further, if any duplication in caching popular contents is removed, then AS will be well provisioned to cache more contents thereby facilitating an additional reduction in traffic over peering links.

In CCN, popular contents are cached over paths connecting clients to the peering link that provides Internet connectivity. Since these paths are not the same for all requesters, popular contents can find themselves cached at different places inside the domain, preventing other less popular contents from being cached as well.

We propose an analytical framework for capturing optimal caching in a CCN domain in Sec. 3.1 and discuss its practical deployment in Sec. 3.2.

3.1 Theoretical Model

Since every non-cached content needs to be fetched from outside the domain, it follows that *on-path caching* is not optimal from peering link bandwidth usage point of view as stated in Proposition 1.

PROPOSITION 1. *The amount of traffic on the peering links of an AS that can cache N contents is minimized if the N top most popular contents are cached.*

PROOF. Under the assumption that each content has the same size, the most popular content (e.g., content of rank 1) accounts for more traffic than any other content. Recursively, any content of rank i accounts for more traffic than any other content of rank $\geq i+1$. Hence, the total traffic accounted for by the N most popular contents is larger than the traffic accounted for by any other combination of N contents. Therefore, as the retrieval of a cached content does not generate

any traffic on peering links, the usage of peering links is minimized for caching the N most popular contents. □

Enforcing exclusively one cache router for each content within an AS needs that every Interest packet for that content must be directed to that router. If the popularity of each content is known, the solution should pass by assigning a particular router to each of the popular contents and to deflect all the Interest packets for this content to that router. However, despite the on-path caching being sub-optimal for bandwidth, it reduces the average content delivery time [6] as popular contents are likely to be cached closer to clients. In order to retain this valuable property of reduced delay, the optimal placement of contents on the different CCN routers must take into account the delay in the AS, such that the selected router for a content is close to clients mostly requesting it. The assignment of routers to contents can be formulated as an integer linear optimization problem with as objective function to minimize the sum of delays over deflected contents. Since we are targeting the optimal behavior, the number of deflected contents is equal to the number of content cache entries available in routers according to Proposition 1 and the number of contents to be cached by a router does not exceed its caching capacity. This optimal caching policy leads to a hit rate close if not equal to one for top-N contents, and to zero otherwise. The sum of delays to minimize for an optimal assignment of routers to contents can be written as follows:

$$\sum_{c\in\mathbf{C}}\sum_{e\in\mathbf{E}}\lambda_{c,e}\sum_{r\in\mathbf{R}}A_{r,c}\cdot d_{e,r}, \tag{1}$$

where $\mathbf{C}$ is the set of deflected contents, and $\lambda_{c,e}$ the rate of Interest packets for content c issued by an edge router e in the set of all edge routers $\mathbf{E}$ (i.e., content popularity). $\mathbf{R}$ is the set of routers with caching capability. The round-trip delay between two routers a and b is given by $d_{a,b}$. The matrix $\mathbf{A}$, composed of all the possible $A_{r,c}$, models the optimal assignment we are looking for. Each entry $A_{r,c}$ is a binary digit in {0,1} that indicates whether router r is assigned to popular content c. From Proposition 1, we must constraint the space of solutions such that each popular content is cached at exactly one router and the caching capacity of each router inside the AS is not exceeded. This gives the following constraints on matrix $\mathbf{A}$,

$$\sum_{r\in\mathbf{R}}A_{r,c}=1,\qquad A_{r,c}\in\{0,1\},\forall c\in\mathbf{C}, \tag{2}$$

$$\sum_{c\in\mathbf{C}}A_{r,c}\leq \text{memory}_r,\qquad \forall r\in\mathbf{R}. \tag{3}$$

If the AS is not over provisioned, the bandwidth on each link l must be constrained to the normalized link capacity c_l as below:

$$\sum_{c\in\mathbf{C}^+}\sum_{e\in\mathbf{E}}\lambda_{c,e}\cdot\delta_{l,e,\text{egress}_{c,e}}\leq c_l \tag{4}$$

where $\delta_{l,a,b}$ is a binary value that indicates whether the link l is on the path between the directed pair of routers $\langle a,b\rangle$, $\mathbf{C}^+$ is the set of all the contents, and $\text{egress}_{c,e}$ is the peering router for content c from edge router e. In order not to overload peering links, the definition of δ must be extended such that, if l is a peering link, $\delta_{l,a,b}$ is verified if b is the peering router to which l is attached. Finally, matrix A can be solved using classical integer linear programming toolboxes.

3.2 Deployment Issues

To implement the above optimal model, we need three mechanisms: (*i*) a way to determine the N most popular contents and their popularity in order to solve the optimization problem, (*ii*) a technique to decide whether to deflect an Interest packet or not, and (*iii*) a way to perform the deflection itself at a CCN router.

3.2.1 Popularity Estimation

In order to track N most popular contents, we propose an algorithm to estimate content popularity. The algorithm maintains a list of $\alpha\cdot N$ counters ($\alpha\geq 1$), each associated to a content name. The value of the counter corresponds to an estimation of the number of interests for the associated content during the observation period. Each time a new content name is observed, a new counter is created and is associated to the name. If the list of counters is full, the entry in the list with the minimum number of tracked requests is removed and a new counter is created to track the number of requests for the new name. In case there are more than one counter with the same minimal value, the one that has been updated the least recently is removed. This management policy ensures that only most popular contents are tracked for their popularity, an information needed for the optimization problem. As our solution captures primarily the request rate along with the freshness of requests among the contents of same rate, thus it can be perceived as combination of LFU and LRU cache management policies. Updates of popularity are of time complexity $\mathcal{O}(log(\alpha\cdot N))$ and memory requirement is of order $\mathcal{O}(\alpha\cdot N)$. To estimate global popularity of contents for an AS, each ingress router executes the above algorithm and sends its popularity estimate to a chosen central router. The central router then performs normalization over all popularity estimates and finally computes the globally N most popular contents.

3.2.2 Decision Process

When an ingress router receives an Interest packet, it checks if the content name in the interest corresponds to any of the N most popular contents, if so, the Interest packet must be deflected. Otherwise, the Interest packet must be forwarded to an egress router in such a way that retrieved content is not cached at the interme-

diate routers on the way back. To determine the router to which the Interest packet must be deflected, the optimization problem is solved centrally. Consequently, each content in the set of globally N most popular contents is tagged with one router that will work as an off-path cache for the content. This optimal assignment of routers needs to be recalculated every time there is an important change in the estimation of globally N most popular contents. The information about globally N most popular contents tagged with corresponding assigned routers is shared with all ingress routers. When the number of routers (and thus matrix A) becomes large, solving the optimization problem could be time consuming. To reduce the size of the matrix, contents can be aggregated into bins with the popularity of each bin equal to the sum of popularities of its contents. The optimization problem can be solved over bins themselves (instead over each content) so as to find an optimal router assignment at the bin level. The formation of bins can be done in different ways, one possible way is to create bins close in terms of number of contents (bins requiring approximately equivalent bandwidth) and overall popularity (to balance load of intra-domain links).

3.2.3 Encapsulation Based Deflection

For each Interest packet corresponding to N most popular contents, the above decision process gives the router working as off-path cache for the requested content. The Interest packet must be encapsulated and forwarded to this router. Knowing that CCN does prefix matching to determine the face to which the Interest packet must be forwarded, a CCN encapsulation can be implemented as follow. Each router, assigned with a unique name, advertises within the AS that it can reach any content with a name starting with its own router name. For example, a router with name `router_id` advertises the prefix `/router_id/`. The name of the content in an Interest packet to be deflected to any given router is then prepended with the prefix associated to this router. The prepending is done only once by ingress router and then the prepended Interest packet can be forwarded as usual in CCN. Now, when a router receives an Interest packet for a content name starting with its own prefix, it strips this prefix, looks for the content in its cache and if found then sends content as Data packet with the same prepended name. If the content is not yet cached, it forwards the Interest packet to a closest peering router to fetch the requested content from outside the AS. The FIB overhead per router remains linear with the number of deflected contents and routers $\mathcal{O}(|\mathbf{R}| + N)$, where $|\mathbf{R}|$ (i.e., number of routers) is due to the fact that each router has to advertise its unique prefix to enable the deflection.

4. CACH: CACHING ALL CONTENTS BY HASHING

The cost of deploying optimal off-path caching can be pivotal as (i) it requires tracking popular contents and solving the optimization problem for the optimal assignment of routers to contents, which can be of non negligible complexity when the number of contents to track and the routers in the AS are large, and (ii) FIB inflation bounded by $\mathcal{O}(|\mathbf{R}|+N)$ can be also high when N is large. To keep the system tractable, we propose *CACH*, a simple heuristic that implements the idea of deflection without the need of solving optimization in Sec. 3. All Interest packets for the same content are forwarded to the same router in CACH. For that purpose, CACH uses a hash function to randomly decide the placement of every content on the routers, independently of the popularity. The hash function is applied to the content name carried in the Interest packet. Routers still need to announce their prefixes for the encapsulation to work, the FIB inflation is then bounded by $\mathcal{O}(|\mathbf{R}|)$ which scales with the size of the AS. CACH simplifies the operation at the expense of a lower hit rate as all the contents are deflected, instead of just the most popular ones, to randomly chosen routers.

5. EVALUATION

In this section, we evaluate the gain in term of bandwidth on peering links usage and discuss the impact of deflection over the content retrieval delay. We built a simulator that uses a given network topology and populates clients that generate Interest packets according to a content popularity distribution being fed to them. We consider four different scenarios for simulations, (i) normal CCN behavior with expected *on-path* caching (in red color in all figures). (ii) optimal off-path caching, where the *optimal placement* of N most popular contents (in green color), on the caches inside an AS is pre-determined. (iii) optimal off-path caching with popularity distribution approximated by the *popularity estimator* discussed in Sec. 3.2 with $\alpha = 1.5$ (in black color), (iv) *CACH* heuristic where hashing is invoked to place or retrieve the popular contents on the fly (in blue color). The simulator is fed with real topologies annotated with latency and IGP weights taken from Rocketfuel [5]. We performed simulations on five such topologies but we only present the results for the ASN 3,967 topology because it is the most challenging among the five topologies as it contains the highest average link latency. This topology is composed of 79 core routers and 44 client routers. The simulator is set-up to randomly choose 44 client routers (two per city) and 6 egress routers that are attached to a peering links. The delay to retrieve a content from a peering link is set to 150 ms. Each core router implements an LRU cache

management policy with a capacity to cache 10 contents resulting in global caching capacity of 790 contents. The choice of small cache size at each core router exacerbates competition between contents to be cached and hence stresses the used caching technique. The total number of contents that can be requested is set to 7,900 so that the AS can globally cache 10% of total number of contents. Each scenario is simulated 11 times and during each simulation 200,000 Interest packets are generated with a content popularity distribution at each client being Zipf (0.8). Request rates at all the clients are considered to be uniform. It is worth noticing that for a given popularity distribution, the total number of contents has no impact as long as the proportion of cacheable contents is constant.

5.1 Peering Link Bandwidth Gain

To determine the peering bandwidth reduction we count the number of Interest packets that leave the AS. Fig. 1 shows the cumulative number of Interest packets that traverse the peering links (i.e., the peering link bandwidth usage) where contents are ordered by popularity. We observe in Fig. 1 that with on-path caching 166,479 Interest packets out of total 200,000 Interest packets (i.e., 83%) have to exit the AS to fetch the requested contents, while for the optimal off-path caching the value decreases to 69,509 (i.e., 35%). The CACH lies between these two extremes with 94,657 Interest packets (i.e., 47%) forced to leave the AS. Fig. 1 also depicts that, with on-path caching 50% of the peering link traffic is generated by top 437 popular contents out of total 7,900 contents (i.e., 5.5%) while the same contents account for less than 0.7% and 22% of peering traffic for the optimal placement and CACH, respectively. Fig. 1 also shows that optimal placement with approximated content popularity distribution is virtually equal to the optimal placement with perfect distribution knowledge confirming close approximation accuracy. As the results obtained with the popularity estimator overlap with that of optimal placement, we assimilate them in the remainder of the paper.

Fig. 2 complements Fig. 1 and gives average hit ratio per content, ordered by popularity (with 95^{th} confidence interval). As anticipated, with the optimal placement the hit ratios for the most popular contents are close to one and zero for the other contents. Comparatively, the hit ratio remains consistently low with on-path caching. With CACH, the hit ratio is high for popular contents and it gracefully degrades with the popularity decrease. Observations show larger hit ratio variance for CACH because there are unpopular contents are also competing for cache resources leading to higher chances of cache eviction.

5.2 Impact of Deflection on Delay

Deflection comes at the cost of increase in delay since Interest packets traverse longer paths in case of off-path caching. However we discover that this inflation in delay is globally compensated by the improved hit rate achieved with off-path caching. Considering traffic composed of Interest packets that are satisfied within AS and ignoring the Interest packets that exit the AS, we observe that the on-path caching offers the lowest average delay ($5.11ms$) whereas the optimal placement gives $23.52ms$ and the CACH gives $28.08ms$ delay. The on-path caching has lowest average delay because it tends to cache the contents closer to clients whereas optimal placement introduces delay due to the deflection. Moreover, optimal placement has better average delay than CACH because it performs optimal placement of content in terms of delay whereas with CACH placement it is completely random. When considering entire traffic (i.e., including Interest packets exiting) the average delay with optimal placement is $84.23ms$ and $154.42ms$ with on-path caching. This is due to the fact that duplication of contents in on-path caching leads to inefficient use of cache space that causes more Interest packets to exit the AS. Moreover, we observe that CACH performs better than on-path caching with $119.19ms$ delay on average.

In addition to global delay reduction, relative gain in delay with respect to contents popularity is necessary to understand traffic dynamics. Fig. 3 shows the average delay (with the 95^{th} confidence interval) of each content, ordered by popularity. Optimal placement consistently provides better delay for popular contents than on-path caching, whereas for unpopular contents, the delay is statistically equivalent. The reason being that in the optimal off-path caching popular contents are always placed optimally for which the delay is minimized. Interestingly, for unpopular contents the delay with on-path caching is similar to that of off-path caching because even in on-path caching unpopular are less likely to be cached in AS (see Fig. 2) due to intensive duplication of popular contents. With CACH, delay for popular contents is better than with on-path caching because larger number of contents are potentially cached. Nevertheless, for the least popular contents, delay is higher with CACH than with the off-path caching because with CACH all Interest packets are deflected. Moreover, variance in delay is higher with the CACH compared to others because placement is random. Table 1 summarizes the delay observations.

Our evaluation shows a clear drop in peering link bandwidth usage as an essence of deploying off-path caching because it improves the hit rate by pruning the duplication in content caching.

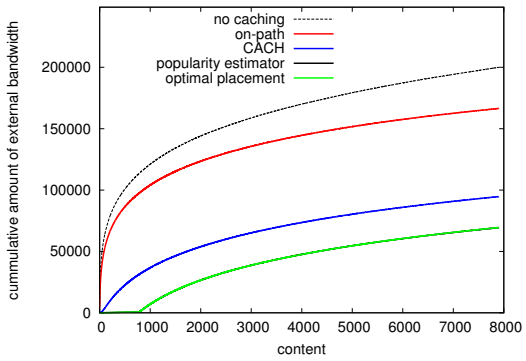

Figure 1: Median of the aggregate traffic on peering links, ordered by popularity

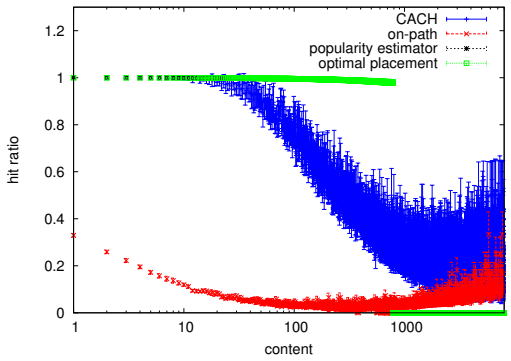

Figure 2: Average hit ratio with 95th confidence interval, ordered by popularity

On-path	CACH	Optimal placement	On-path	CACH	Optimal placement
$5.11ms \pm 0.05$	$28.08ms \pm 0.04$	$23.52ms \pm 0.03$	$154.42ms \pm 0.05$	$119.19ms \pm 0.11$	$84.23ms \pm 0.09$

Table 1: Summary of the average content retrieval delay. Left part for content effectively in a cache, right part accounting all contents.

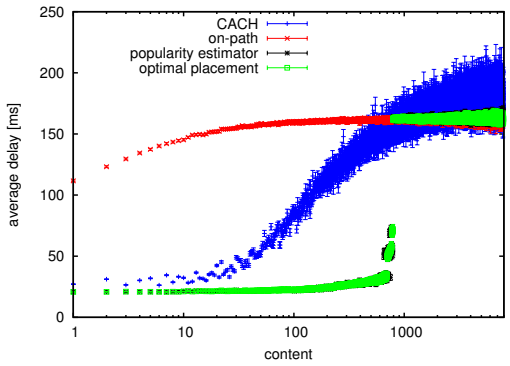

Figure 3: Average delay with 95th confidence interval, ordered by popularity

6. CONCLUSION

The existing CCN architecture is a future networking paradigm to cope with alarming drift in today's Internet usage. In particular, CCN advocates opportunistic caching at each router on the path from content source to content consumers. We prove that this caching technique is not optimal in terms of peering links' bandwidth usage as it results in wastage of limited caching space by duplicating contents throughout the network.

In this paper, we propose a way to deploy an effective caching within an Autonomous System (AS) such that peering link usage is minimized Our optimal off-path caching solution needs to deflect selectively the Interest packets corresponding to the cached popular contents towards the optimally elected off-path routers (caches). We demonstrate with simulations on real topologies that the cost of using off-path caching in terms of delay due to the deflection is compensated by the gain in terms of caching. Indeed, as caching is optimized, a larger number of popular contents can be cached within the AS and hence less traffic exits the AS via the peering links which results in an overall delay reduction. Further, we propose a heuristic named CACH as a practically implementable caching solution with low overheads. We demonstrate by simulations that the CACH approximates well the optimal off-path caching.

7. REFERENCES

[1] C. Fricker, P. Robert, and J. Roberts, *A versatile and accurate approximation for LRU cache performance*, arXiv, 2012.

[2] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, *Networking named content*, CoNEXT '09, ACM, 2009.

[3] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. Kim, S. Shenker, and I. Stoica, *A data-oriented (and beyond) network architecture*, SIGCOMM '07, ACM, 2007.

[4] D. Lagutin, K. Visala, and S. Tarkoma, *Publish/subscribe for internet: Psirp perspective*, Future Internet Assembly, 2010.

[5] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, *Inferring link weights using end-to-end measurements*, Proc. of the 2nd ACM SIGCOMM Workshop on Internet measurement, IMW '02, ACM, 2002.

[6] L. Muscariello, G. Carofiglio, and M. Gallo, *Bandwidth and storage sharing performance in information centric networking*, Proc. of the ACM SIGCOMM workshop on Information-centric networking, ICN '11, ACM, 2011.

[7] N. Niebert, S. Baucke, I. Khayat, M. Johnsson, B. Ohlman, H. Abramowicz, K. Wuenstel, H. Woesner, J. Quittek, and L. Correia, *The way 4ward to the creation of a future internet.*, PIMRC, IEEE, 2008.

[8] N. Spring, R. Mahajan, and D. Wetherall, *Measuring isp topologies with rocketfuel*, Proc. of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM '02, ACM, 2002.

[9] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. Thornton, D. Smetters, B. Zhang, G. Tsudik, kc claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh, *Named Data Networking (NDN) Project*, www.named-data.net, 2010.