Exploit the known or explore the unknown? Hamlet-like doubts in ICN

R. Chiocchetti, D. Rossi, G. Rossini
Telecom ParisTech, Paris, France
first.last@telecom-paristech.fr

G. Carofiglio, D. Perino
Alcatel Lucent Bell Labs, Nozay, France
first.last@alcatel-lucent.com

ABSTRACT

Most Information Centric Networking designs propose the usage of widely distributed in-network storage. However, the huge amount of content exchanged in the Internet, and the volatility of content replicas cached across the network pose significant challenges to the definition of a scalable routing protocol able to address all available copies. In addition, the number of available copies of a given content item and their distribution among caches is clearly impacted by the request forwarding policy.

In this paper we gather initial design considerations for an ICN request forwarding strategy by spanning over two extremes: a deterministic *exploitation* of forwarding information towards a "known" copy and a random network *exploration* towards an "unknown" copy, via request flooding. By means of packet-level simulations, we investigate the performance trade-offs of exploitation vs exploration approaches, and introduce an hybrid solution. Our forwarding scheme shows a good potential in terms of delivery performance, implicit cache coordination and possible reduction of forwarding table size.

1. INTRODUCTION

Information Centric Networking (ICN) introduces a new networking paradigm, where the communication is centered around named-data, rather than host addresses. Indeed, in ICN every content item is identified, addressed and retrieved by its name instead of its physical location. All network nodes potentially store the data they forward to serve future requests for the same content: this feature may help reducing the transport cost for network providers and improving end-user delivery performance.

To this aim, most ICN designs [2] propose to equip network nodes with enhanced storage capabilities. Storage resources are used to maintain temporary content replicas spread through the network for a period of time ranging from minutes to hours or days. The availability of different replicas depends on several factors, like content popularity, cache replacement policies, and so on, and is clearly impacted by the request forwarding policy.

An ideal name-based routing protocol would have to address all temporary copies of every content item in order to forward user requests towards the "best" available replica (e.g., closest). However, this is clearly unfeasible in ICN for different reasons:
- the *network scale*; ICN paradigm applies to content of different applications and is not intended to be confined to small, controlled network regions;
- the *time scale*; temporary copies stored at network nodes are highly volatile and the signaling overhead involved by frequent route updates would be excessive;
- the *forwarding table (FIB) size*, that is already a matter of concern within all ICN designs, even considering only *permanent* content copies rather that network-cached temporary replicas [13].

In this paper, we tackle the problem of the definition of a scalable forwarding policy of content requests suitable for ICN, focusing on an intra-domain point of view. Our objective is to design a forwarding strategy that: (i) can discover temporary content replicas and forward requests accordingly; (ii) requires little or no a priori knowledge; (iii) does not generate additional signaling overhead; (iv) can achieve implicit cache coordination.

To this purpose, we consider a range of solutions spanning over two extreme strategies: a deterministic *exploitation* of forwarding information towards a "known" permanent copy of the requested object, vs a random network *exploration* towards "unknown" cached copies, via request flooding. The first approach reduces the amount of requests that need to be spawned over the network – though, as requests follow paths toward permanent copies, they may miss even closer cached replicas along other paths. Conversely, the second approach has the potential of reducing the FIB size and the signaling overhead generated by frequent route updates, but it does not guarantee optimal delivery performance.

By means of packet-level simulations, this paper investigates the potential benefits and performance tradeoffs of request forwarding solutions based on exploitation versus exploration. Further, we introduce an hybrid request forwarding scheme designed to take advantage of both approaches.

2. BACKGROUND

A large body of literature focuses on inter-cache coordination and on routing of content requests, as tightly coupled with cache coordination or content placement. Besides architecture and technology-specific constraints, such algorithms can be divided in two broad categories: explicit and implicit schemes.

Explicit schemes [9, 3, 5, 6, 12, 17] require a perfect knowledge of key system parameters like network topology, content popularity, storage capacity and request routing.

The joint problem of object placement and request routing in a CDN is addressed in [3]. Two optimal algorithms are provided for replicating content on the proxy servers and for routing content requests to a suitable server, minimizing the total distribution cost.

A random hashing scheme is proposed for tree-like cache networks in [9]. According to the authors, the main drawback of this technique consists in the explosion of the number of messages between caches, as the size of the network grows, even with multicast.

Authors in [5] mainly develop an optimal content placement policy for an IPTV system in a hierarchical tree-like topology, where requests are routed from the user to upper cache levels, until the first replica of the content is reached. Unlike [5] and classical content placement work, [6] focuses on joint content placement and dynamic request routing to realize a distributed cache cooperation for IPTV service. Along the same line, [12] develops a cooperative caching strategy for on-demand IPTV streams with the aim of minimizing cross-domain traffic by jointly addressing content placement and routing in CCN.

Explicit schemes have the advantage of achieving an (nearly) optimal distribution of content replicas and of content requests by leveraging network knowledge. However, this is done at the expense of a considerable overhead in terms of the amount of state required, and to the number of messages that have to be exchanged to maintain and to update such state. For these reasons, such schemes may be unfeasible in the context of ICN, mainly due to the large scale and the highly dynamic content demand.

Implicit schemes[15, 11, 14, 4, 16] realize cache coordination and request routing without continuous exchange of information among nodes, and are thus more suited for ICN. Different criteria, as listed in [15], can be used for mapping content requests to nodes storing a local replica: static priority, minimum load, fastest response, round-robin, random split, hash based.

Additionally, [15] provides a name-based mapping of content request to server by using a random hashing function to bind a content to one of the servers storing a local replica. Although hash-based scheme do not require regular message exchange between caches, the mapping (hence, the routing) is static.

An Informed Caching Proxy Selection technique is developed in [11], which allows to infer remote cache properties by observations of its miss stream. Optimal request forwarding is achieved with no message exchange between caches. However, the amount of state that has to be kept at each cache for inference purposes is not negligible.

In [14], authors propose a best effort content location based on "breadcrumbs", that is pieces of information (kept at each node) about the history of content requests. This method allows to track the evolutions of the request pattern, at the price of per-object soft-state information.

In highly distributed cache systems, load-aware source selection strategies have been shown to perform better than load-oblivious server selection. Recently, [4] compares random selection and Shortest-Queue selection over the same number of neighboring nodes, to conclude that load-oblivious random policies lead to an increased retrieval latency in presence of a large number of edge nodes. On the contrary, load-aware Shortest Queue selection appear to sustain the scalability of the content distribution system.

In mobile ad-hoc networks [16], request routing by flooding is preferred due to the redundancy in forwarding necessary to compensate the unreliable nature of the medium. However, plain flooding of content requests is shown to be hardly viable and "mitigated flooding" is defined where requests have limited temporal validity and propagate within a certain radius. The selection of these parameters is however critical for a correct and efficient system operation.

Differently from previous work, this paper aims at combining request forwarding based on information stored in the FIB, built for instance using explicit schemes over long timescales, with request flooding, used to discover temporary copies without explicit signaling and significant state to maintain at every node.

3. SYSTEM DESIGN

Rather than selecting a specific ICN architecture *a priori*, we present an high-level overview of the algorithmic design space for request forwarding policies, after which we evaluate some representative candidates.

3.1 Design space

The design space we consider in this work is summarized in Tab. 1. We observe that all ICN architectures exploit a similar mechanism, i.e., users *express requests* to *retrieve given data* [7]. Requests are forwarded by network nodes until one copy of the desired content is found, and in-network storage is exploited to tempo-

Table 1: ICN system design space

FIB Knowledge	• None • Omniscient • Partial
Request forwarding strategy	• Exploration • Exploitation • Hybrid
Data unit	• Monolithic object-level • Partitioned chunk-level

rary cache forwarded data. We argue that the strategy according to which requests are forwarded (i.e., along which routes requests are sent) can heavily influence the distribution of content cached in the network (and the delivery performance perceived by users as a consequence).

The *request forwarding strategy* is constrained by the information available in the forwarding table (FIB). In case ICN nodes have at their disposal FIB information useful to forward requests towards one (or more) server where a persistent copy of the data is stored (e.g., the shortest path to a given content originator), this *information can be exploited* by the ICN architecture. Content can be found only on "en-route caches" (i.e., caches between the content requester and the content originator), possibly failing to find nearby cached copies (e.g., that lie along the shortest path of another close requester). Similarly, data will be cached on the path between the content originator and the requester only.

On the other extreme, *in case no useful FIB information is available, then the neighborhood need to be explored* to find a temporary copy. In this case, requests are expressed over possibly multiple paths, which can lead to the usual drawbacks of flooding-based algorithms (and require the usual counter-measures, like TTL-based scoping or probabilistic pruning of some branches in the exploration process). Similarly, temporary copies will then be available at multiple neighbor nodes.

Hybrid strategies can be devised when nodes have only partial knowledge of existing routes towards available content items, or in order to exploit nearby temporary replicas. Indeed, given the large amount of content available in the network and the volatility of cached copies, it is not feasible to design a scalable routing protocol able to address all available copies of every item. Accordingly, only part of the requests are deterministically sent over "known" routes, while the others are forwarded via flooding schemes to find "unknown" copies.

Yet another dimension that may draw the exploitation vs exploration design is the *minimal data unit*. At two antipodean extremes, content of requests can be either monolithically requested by users and cached by intermediate routers (*object-level*) or partitioned in chunks (*chunk-level*). Clearly, there is a deterministic overhead when ICN is used in chunk-mode, as multiple requests have to be expressed to retrieve the same data. At the same time, requests are generally of limited size, and chunk-mode offers a greater flexibility for data retrieval. Furthermore, requests for the first chunk of any given object could be expressed according to an *exploration* paradigm: this would lead to the discovery of the path toward the closest (cached) object copy, that could be stored in a soft-state FIB cache and be *exploited* by requests for subsequent chunks of the same object.

3.2 Reference strategies

Based on the previous discussion, in this paper we assess pros and cons of the above families of approaches. As our assessment involves both *qualitative* as well as *quantitative* observations, we define and implement a few representative strategies for each family.

Exploitation. Nodes have FIB knowledge concerning the placement of the (possibly multiple) permanent copies of any object in the system. In case multiples original copies are stored in the system, a (uniformly) randomized selection of the server toward which requests are sent is performed. Furthermore, the selection is randomized for each new chunk request in case of chunk-mode ICN architectures. Under the exploitation paradigm, while the number of messages and requests sent is possibly lower, FIB management (i.e., routing, lookup) may have a non-negligible cost, and caching can only happen en-route in the backward path to the originator (so that possibly closest cache is not found with this strategy).

Exploration. Nodes have no FIB knowledge and are forced to flood requests. Instead of a fixed-scope limit (e.g., by requiring the number of hops n to be $n < TTL_{max}$), we limit the flooding scope probabilistically, such that at the n-th hope, the request message is flooded with exponentially fading probability β^n (as typical in reinforcement learning). Additionally, we *aggregate* requests at nodes (as done in [8, 10]), to avoid sending subsequent requests when a first outstanding request for the same object has not been satisfied yet. As previously introduced, the network is explored only for the first chunk in the case of chunk-mode partitioning. Under exploration, closest copies are expected to be found, at the expense of a more intense communication, that is however possibly limited to the first chunk.

Hybrid. Nodes may have partial FIB knowledge that allows them to forward requests in an exploitation-based approach, whereas the rest of the catalog will be served by an exploration-based scheme. In more details, since it is likely for popular content to be stored in nearby caches, exploration can be used for the first $\mathcal{K}$th percentile of the request distribution. Instead, for the remaining part of the catalog, the hybrid strategy exploits FIB knowledge (since least popular content it may be necessary to go up to the content originator). Notice that the hybrid case, $\mathcal{K}$ acts as a cutoff parameter, smoothly tuning between pure exploitation ($\mathcal{K}$= 0) or exploration ($\mathcal{K}$= 1).

3.3 Practical consideration

On the one hand, routing and forwarding represent a possible scaling bottleneck for exploitation-based ICN architectures. The problems are that (i) advertisement[1] of content name may require significant control load with respect to IP address ranges (ii) ICN nodes may keep very large routing tables so that the FIB table lookup may be prohibitively complex. As such, exploration-based techniques may assist in reducing the FIB size to a more manageable amount.

On the other hand, while exploitation is guaranteed to converge to at least a permanent copy (i.e., the one stored by one of the repositories), this is no longer the case under the exploration paradigm. Furthermore, in case the exploration scope is too tight, requests may not even reach an ICN node holding a cached copy of the content of interest. As such, exploration-based techniques need to re-express unsuccessful requests after a timeout. As for the time being we are not investigating user-performance, in this work we set a conservative timeout RTO equal to the RTT delay between the two most faraway nodes in the network.

Finally, some remarks are necessary concerning the cutoff $\mathcal{K}$ parameter for the hybrid case. First, we point out that before devising practical mechanisms to implement hybrid strategies, it is necessary to assess whether there would an be actual interest (e.g., significant performance gain) for their implementation – which is precisely the purpose of the present investigation. Second, the cutoff could be *implicitly* implemented by disseminating only a portion of the FIB entries in the routing protocol, which would require a (possibly rough) estimation of the content popularity – that can be practically implemented at manageable cost.

4. PERFORMANCE EVALUATION

We implement the above strategies in a custom C++ simulator[1] using the Omnet++ framework. Though our aim is to gather initial insights on the above approaches (more than providing their thorough performance evaluation, which we leave as a future work), we however take great care in defining a relevant scenario.

We consider a real topology, namely the well-known Geant network (comprising 22 nodes and having a diameter of 6 hops). We assume to operate in a non-congested regime, and consider infinite capacity links. Additionally, for the sake of simplicity, we consider homogeneous network settings where each backbone link has the same propagation delay equal to 1 ms (so that we set exploration $RTO = 12\,\text{ms}$). Clients aggregate are attached to each backbone node, while content repositories are located on a few Point of Presence (PoPs). In more detail, each PoP is attached to a different backbone ICN node, and we consider a number of servers varying in $\|PoP\| \in \{1, 4\}$.

We consider for the time being a catalog comprising $N_{obj} = 10^4$ individual objects, having geometrically distributed size with a mean of 10 MBytes. In the case of chunk-mode ICN architectures, we consider that chunks have 100 KByte size, so that on average each object consists of 100 chunks. The aggregate request stream for an object i arrives to each ICN node with a rate $\lambda_i = p(i)\lambda, i \in F$, where $p(i)$ expresses the content popularity for object i. In this work, we consider $p(i) = \frac{C}{i^\alpha}$, $C = \left(\sum \frac{1}{i^\alpha}\right)^{-1}$, i.e., popularity follows a Zipf distribution, with exponent $\alpha = 0.9$. By definition, for each ICN node the aggregate request rate over all objects is $\lambda = 1\,\text{Hz}$.

Notice that while exploitation do not rely on any parameter, exploration performance depends on the backoff parameter β, while hybrid performance depends on the cutoff $\mathcal{K}$. In this section, we first assess exploration sensitivity to β for object-mode. We then compare the three techniques for object vs chunk-mode, under both single and multi-PoP scenarios.

4.1 Object-mode Exploration

Exploration dynamics are reported in Fig. 1 as a function of β. We report the request and data cost on the left y-axis, and the time it takes a request to hit a cache on the right y-axis. For each request expressed by any given source, the request cost counts the number of time that request has crossed a network link. Similarly, data cost expresses the number of links crossed by data generated in reply to user requests. Notice that these *cost* metrics simultaneously weight two factors: (i) the number of requests/data traveling in the system as well as (ii) the distance they have traveled. Notice also that due to the difference between request vs data messages size, request cost is tied to the *processing complexity* of the ICN system, while the data cost expresses the *network load*. Finally, we use the time it takes a request to hit a cache on the right y-axis to ensure the correctness of the design, more than an expression of the user performance (due to uniform link delay and infinite ca-

[1] Aggregation of content names prefixes is expected to be less efficient with respect to IP address space aggregation. This is surely the case for flat identifiers and also likely for hierarchical DNS-like names (recall that the fanout for the *.com* top-level domain alone is on the order of 100,000,000 domains).

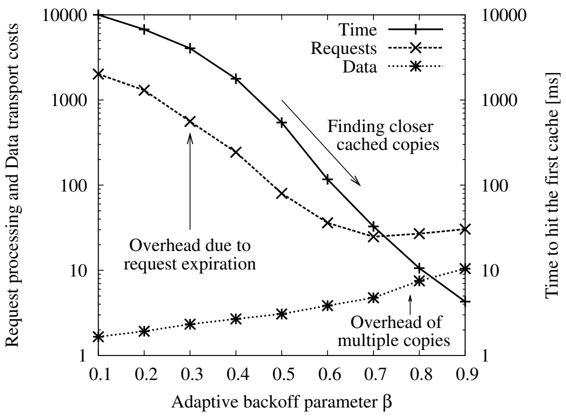

Figure 1: Exploration: request processing and data transfer costs (left y-axis) and time duration of a lookup (right y-axis) as a function of the backoff parameter β under object-mode architectures.

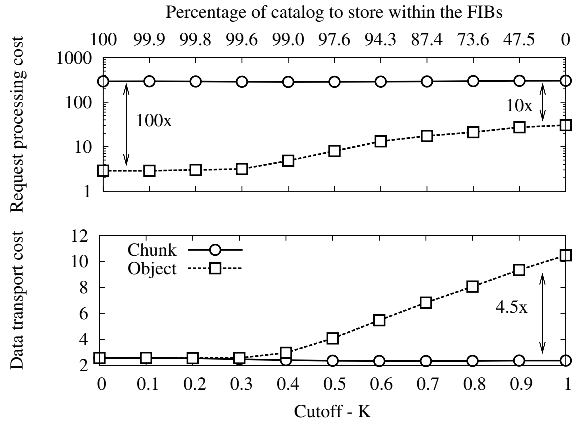

Figure 2: Object- vs chunk-mode comparison of exploitation ($\mathcal{K}$= 0) vs exploration ($\mathcal{K}$= 1) vs hybrid for (varying $\mathcal{K}$).

pacity).

First, notice that when β is low, exploration is likely not successful, so that timeouts will often trigger new requests for the same content. This in turn leads to a high request load, and to a very large delay before a cache is hit. Basically, requests are successful only provided that previous requests have already let the content to be cached in a nearby ICN node; hence, successful requests slowly propagate from nodes closer to the content repository to faraway nodes.

As β increases, the number of retransmissions decreases, as requests explore a wider network portion, and are more likely to hit a cache. At the same time, as a result of the higher cache hit probability, the number of copies received possibly increases as well (exponentially with the backoff parameter: notice the logarithmic y-axis scale). Yet, as requests are aggregated at ICN nodes, no broadcast storm ever happen, so that the system is stable even for $\beta \to 1$.

As a result of this preliminary investigation, we set $\beta = 0.9$ in the reminder of this work as a good compromise between (i) limited data cost, (ii) low request rate and (iii) low delay.

4.2 Object- vs Chunk-mode Comparison

We now consider the three strategies early outlined. Recall that exploitation ($\mathcal{K}$= 0) and exploration ($\mathcal{K}$= 1) are special cases of the hybrid strategy. We investigate the system performance as a function of the cutoff parameter $\mathcal{K}$ in Fig. 2. The bottom x-axis represent the percentage of the Zipf request $\mathcal{K} = \sum_{j=1}^{k} 1/\alpha^j$, while the top x-axis represent the percentage of the catalog k/N_{obj} corresponding to the $\mathcal{K}$ percentile of requests. In other words, the top x-axis report the percentage of FIB entries that are necessary for any given cutoff $\mathcal{K}$ (assuming 1 FIB entry per object).

Clearly, increasing $\mathcal{K}$ from 0 (full exploitation) to 1 (full exploration), we see that different behavior arise due to object- vs chunk-mode operation. Recall that in case of chunk-mode, flooding is performed only the first chunk, while subsequent chunks may exploit soft-state cached FIB information in return of the first data chunk (as such, object-mode performance are also representative of the cost for the first chunk).

Clearly, as far as request processing cost is concerned, chunk-mode may produce as much as 100× more requests per object (under exploitation). At the same time, as $\mathcal{K}$ grows and so the use of exploration, in object-mode there is a larger overhead per each request (since each request has to be flooded). Conversely, the overhead in chunk-mode reduces, since subsequent requests have a lower individual cost (since only 1/100 has to be flooded and 99/100 follow the exploited path). Hence, the processing cost gap per single request reduces quite significantly (by an order of magnitude).

However, chunk-mode flexibility payoff is even more evident in the case of data transport cost. Clearly, notice that under exploitation, practically no difference arise between object- vs chunk-modes (where we neglect the overhead in terms of packetization and extra headers for the sake of simplicity). Instead, as $\mathcal{K}$ grows and so the use of exploration, the whole object disseminates over multiple links in object-mode, while wide-scope dissemination affect only the first-chunk in chunk-mode.

Overall, in the considered scenario the average request processing cost to disseminate a complete object in chunk-mode is about 10× higher than in object-mode. At the same time, the average data transport cost is about 5× lower, so that these two techniques

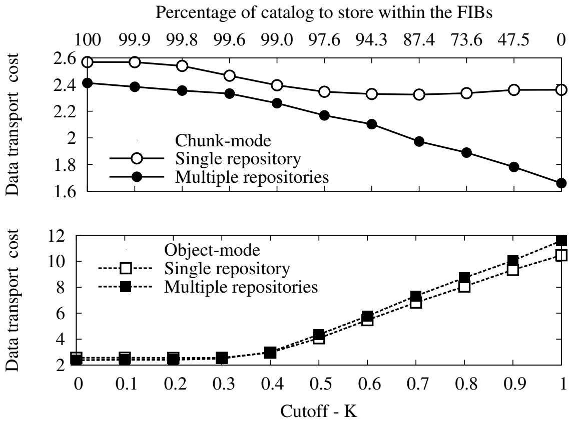

Figure 3: This figure shows different behaviour considering chunked objects (top), and entire objects (bottom) in the case of multiple repositories.

sits at an opposite side of the tradeoff. Finally, notice that, when $\mathcal{K} = 0.9$ the FIB size is cut by a factor of $2\times$: while this may not be enough, Fig. 2 shows the system to be stable when $\mathcal{K} \to 1$, so that it may be possible to further reduce the FIB size[2].

4.3 Single- vs multi-PoP Comparison

We finally consider the case where possibly several persistent object copies exist, that are stored at different PoPs. Exploitation selects one of these PoPs from its FIB: interestingly, in object-mode the data transport cost in case of multiple PoPs may *increase* with respect to the single repository case, as a consequence of a reduced level of aggregation (i.e., many different clients behind the same ICN router can possibly request the same object to different repositories, with a data traveling along a longer path than in the previous case).

In the exploration case instead, since many more alternatives are explored, the odds to find a closer cache are higher in the multiple PoP case. Indeed, as only the first chunk of each object undergoes a full exploration, unpopular contents will not evict the popular one from caches, mitigating the otherwise negative effect of the exploration.

5. CONCLUSION

This paper compares two antipodean approaches, namely *exploitation* vs *exploration*, for request forwarding in ICN networks, focusing on the intra-domain case.

Initial results show that: (i) exploration can improve end-user delivery performance as it allows to discover close content replicas; (ii) a chunk-level exploration of neighboring node caches is more effective than an object level request flooding; (iii) our hybrid forwarding solution is more effective in presence of multiple repositories (i.e. permanent copies of a content); (iv) hybrid strategies can reduce the amount of FIB information that needs to be stored, at the cost of a moderate increase of the number of requests handled by each node.

In our future work we aim at extending these results by exploring a wider design space, including the joint design of request forwarding and cache decision policies, to achieve implicit and efficient coordination of a network of ICN caches.

[2]This depends on the catalog size F_{obj} and Zipf exponent α; it is not the purpose of this paper to precisely quantitatively assess the reduction for more realistic catalogs.

6. REFERENCES

[1] ccnsim homepage. http://www.telecom-paristech.fr/~drossi/ccnSim.
[2] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. A Survey of Information-Centric Networking. In *Dagstuhl ICN Seminar*, 2011.
[3] T. Bektaş, J.F. Cordeau, E. Erkut, and G. Laporte. Exact algorithms for the joint object placement and request routing problem in content distribution networks. *Comput. Oper. Res.*, 35(12):3860–3884, 2008.
[4] M. Bjorkqvist, L. Chen, M. Vukolic, and X. Zhang. Minimizing retrieval latency for content cloud. In *IEEE INFOCOM*, 2011.
[5] S. Borst, V. Gupta, and A. Walid. Distributed caching algorithms for content distribution networks. In *IEEE INFOCOM*, pages 1478–1486, 2010.
[6] J. Dai, Z. Hu, B. Li, J. Liu, and B. Li. Collaborative hierarchical caching with dynamic request routing for massive content distribution. In *IEEE INFOCOM*, 2012.
[7] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox. Information-centric networking: seeing the forest for the trees. In *ACM HotNets*, New York, NY, USA, 2011. ACM.
[8] V. Jacobson, D. Smetters, J. Thornton, M. F. Plass, N. Briggs, and R. Braynard. Networking named content. In *Proceedings of ACM CoNEXT*, Rome, Italy, 2009.
[9] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the world wide web. In *ACM STOC*, pages 654–663, 1997.
[10] T. Koponen, M. Chawla, B-G. Chun, A. Ermolinskiy, K.H. Kim, S. Shenker, and I. Stoica. A Data-Oriented (and Beyond) Network Architecture. In *ACM SIGCOMM '07*, Kyoto, Japan, Aug. 2007.
[11] N. Laoutaris, G. Zervas, A. Bestavros, and G. Kollios. The cache inference problem and its application to content and request routing. In *IEEE INFOCOM*, 2007.
[12] Z. Li and G. Simon. Time-shifted tv in content centric networks: The case for cooperative in-network caching. In *IEEE ICC*, 2011.
[13] A. Narayanan and D. Oran. Content Routing using Internet Routing Protocols: Can it scale? IETF-82, ICNRG BAR BOF, November 2011. URL: http://trac.tools.ietf.org/group/irtf/trac/wiki/icnrg.
[14] E. J. Rosensweig and J. Kurose. Breadcrumbs: Efficient, best-effort content location in cache networks. In *IEEE INFOCOM*, 2009.
[15] D. Thaler and C.V. Ravishankar. Using name-based mappings to increase hit rates. *IEEE/ACM Trans. Networking*, 6(1):1–14.
[16] M. Varvello, I. Rimac, and U. Lee. On the design of content-centric manets. 2011.
[17] Y. Wang and K. Lee. Advertising cached contents in the control plane: Necessity and feasibility. In *IEEE INFOCOM, NOMEN Workshop*, 2012.