# Cache-Aware Routing in Information-Centric Networks


Vasilis Sourlas
Computer & Comm. Engineering,
Univ. of Thessaly, Greece.
Email: vsourlas@inf.uth.gr

Paris Flegkas
Computer & Comm. Engineering,
Univ. of Thessaly, Greece.
Email: pflegkas@inf.uth.gr

Leandros Tassiulas
Computer & Comm. Engineering,
Univ. of Thessaly, Greece.
Email: leandros@inf.uth.gr



***Abstract*—Information-Centric Networking (ICN) has attracted the attention of the research community, which has argued that content, instead of end-points, must be at the center stage of attention. The research issues addressed by most of the proposed architectures are related to persistent/unique naming, efficient content distribution, in-network caching and security. Despite the emergence of ICN-oriented solutions, little attention has been given on designing efficient routing mechanisms suitable for ICNs since most of the approaches assume either traditional shortest path or inefficient flooding schemes. In this paper, we first describe how a resource management system can be deployed on two of the most fermentative ICN network architectures and we propose a cache-aware routing scheme that calculate the paths with the minimum transportation cost based on the information item demands and the caching capabilities of the network. We present a Dynamic Programming approach for calculating the paths and we validate the proposed approach through simulations.**


## I. Introduction

Information-Centric Networking has recently attracted significant attention and several research groups proposed and developed direct information-centric routing architectures, using location independent content ID/names instead of endpoint-node addresses; see NDN [1], PURSUIT [2], COMET [3] and SAIL [4]. Naming content directly enables the exploitation of in-network caching, in order to improve delivery of popular content. Each information item can now be uniquely identified and authenticated without being associated to a specific host, making in-network opportunistic caching one of the salient characteristics of ICN architectures [5]. In this approach every node of the network caches every item that traverse it, and if a matching request is received while an item is still in the cache of a node, it will be forwarded to the requester from that node, avoiding going all the way to the hosting server.

Given this emergence of ICN-oriented solutions, the relevant management needs in terms of performance have not been adequately addressed, yet they are absolutely essential for relevant network operation and crucial for the ICN approaches to succeed. Performance management and traffic engineering approaches are also required to control routing, to configure the logic for replacement policies in cache stores and to control decisions where to cache, for instance. While quite a lot of studies have been performed related to caching, routing functionality is completely missing from the current ICN design with only simple flooding or OSPF-like shortest path mechanisms having been proposed. This choice has been deliberately left open in order to allow routing solutions ranging from schemes potentially based on known protocols to innovative solutions best suited to the specific communication model of ICN.

In-network caching in ICN exhibits fundamental differences from the traditional network caching schemes and poses new challenges [6], [7]. In general, ICN enables caching of addressable information items [5], [8] in every cache-equipped node [9] and replacement of cached items at line-speed [10]. The cache-everything-everywhere scheme presented in [5] has already raised doubts and some authors have already questioned this aggressive strategy [11]. In that direction in [12] authors instead of caching the same item at every node along the delivery path investigate if caching only in a subset of node(s) along the delivery path can achieve better performance in terms of cache and server hit rates by exploiting the concept of betweenness centrality. Nevertheless, every research attempt regarding in-network caching in ICN, takes as granted the use of the shortest (in terms of hops) path as the request path and consequently the delivery path from the requested node to the server, without taking into consideration the caching capability of the used path and the possibility of using an alternative routing scheme.

In the area of routing in ICN authors in [13] present what inter-domain routing policies could look like in an NDN Internet (policies are realized by tuning a variety of parameters or *knobs*, available in the Internet BGP routing protocol). They describe the knobs available to network operators and their possible settings and they explore new economic incentives that may be present in an NDN Internet and consider what types of routing policies they may develop. In [14] authors propose a scalable routing and name resolution framework, called *Scalable Multi-level Virtual Distributed Hash Table* (SMVDHT). SMVDHT uses a combination of name aggregation and multilevel virtual DHTs to achieve scalability. SMVDHT also exploits underlying intra- and inter-domain IP routing protocols to build multi-level virtual DHTs for name resolution, which is more efficient than conventional hierarchical DHT schemes and simplifies network management. Finally, authors in [15] propose the *Potential Based Routing* (PBR) in ICN to achieve several design goals such as availability, adaptability, diversity, and robustness. PBR uses a similar idea with the directed diffusion [16] in a sense that a data packet navigates a potential field defined in the network to facilitate routing towards a desired location. The proposed PBR is more close to the service discovery mechanism [17] which has been applied to mobile ad-hoc networks.

In this paper, we propose a new cache aware intra-domain

978-3-901882-50-0 ©2013 IFIP

routing scheme that dynamically computes the routes followed by each request for each item and from each node in the network. Particularly, we present a dynamic programming (DP) approach for the computation of the cheapest transportation paths based on the observed item request patterns, such as their popularity and locality and the used caching scheme, in order to minimize the overall transportation cost imposed by the user requests. Finally, we present a resource management system architecture for the cache aware routing in an ICN, where resource managers make route decisions, and we validate the proposed approach through simulations.

The rest of the paper is organized as follows. In Section II, we present the functionality of the resource management architecture that will take care the cache aware routing decisions in ICN based on the selected caching scheme and the volatility of the user requests. In Section III, we formulate the cache aware routing problem and present the DP approach for the computation of the cheapest transportation paths given the cache hit ratio of each item at each node of the network. Finally, in Section IV we evaluate the performance of the proposed algorithm and we compare its outcome with the traditional shortest path routing scheme, while in Section V we conclude our paper and give pointers for future work.

## II. ICN Resource Management System Architecture

Despite the fact that in-network caching has emerged as one of the most important research fields in the context of ICN, most of the research attempts assumed the use of the shortest (in terms of hops) path as the delivery path and tried to optimize the network performance through the exploration of various caching schemes in the nodes, without exploring at the same time the caching capabilities of other possibly longer paths. Consequently, efficient management of such networks entails managing both the routing and the caching scheme used at each node with objectives such as minimizing the content access latency from clients, maximizing the traffic volume served by caches and thus minimizing the bandwidth cost and congestion of the server.

In this section, we present an ICN resource management system architecture for controlling the routing processes as well as the cache resources of an ICN network. In Fig. 1, we depict how such a system can be deployed on top of a PSIRP/PURSUIT-like network architecture following a centralized approach while, a distributed approach applied on a NDN-based network is shown in Fig. 2.

In PURSUIT [2], the design paradigm involves three separate elements and three separate functions: publishers, subscribers, and the REndezvous NEtwork (RENE) on one hand, and the functions of rendezvous, topology management/formation and forwarding on the other hand respectively. Although, the PURSUIT architecture originally followed a PUSH communication model, in order to support opportunistic in-network caching a PULL model has also been described in [10]. In this ICN implementation, the entities that decide on the routing and caching scheme adopted in the managed network are called Topology Manager (TM) and Cache and Route Manager (CM) and reside either in a node of the network or in a management server and can be co-located

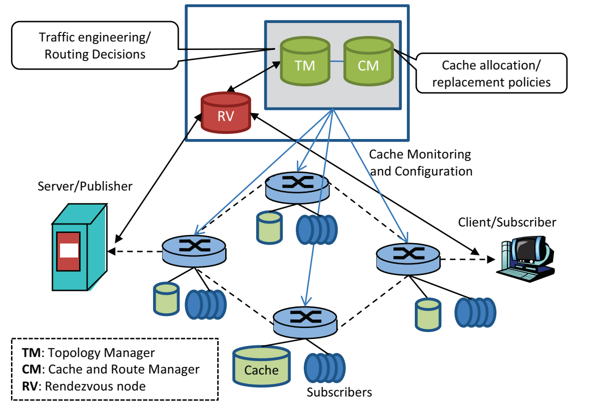


Fig. 1. ICN resource management in a PSIRP/PURSUIT-like network architecture.

with the Rendezvous Node. The TM/CM may either take long term traffic engineering decisions based on predicted information item demand or dynamically control the routing and cache resources based on real-time network information by monitoring the status of the network e.g. cache hit ratios, link utilization, etc. The TM/CM would extract the demand patterns by monitoring the RV node, which gathers the subscriptions of every user in the network in order to bind the publishers to the subscribers, and passes the successful bindings to the TM to compute the paths based on its optimization objectives from the subscribers to the publisher(s). The TM then encodes the path in a Bloom filter and sends it to the subscriber (through the RV) to start sending the requests towards the publisher. In every node the request traverses a backward path bloom filter is calculated and included in the request packet in case an item is found in an intermediate node to be able to find its way back to the subscriber.

In NDN [1] when a user wishes to receive data, he/she issues an Interest packet that contains the data name. The network propagates the Interests to the nearest data source (anycast) and then the requested item is delivered back to the user in the form of a Data packet. Each node uses the *Pending Interest Table* (PIT) to keep track of the forwarded Interests. In Fig. 2, a distributed approach is depicted, where a Resource Manager (RM) is installed in every node of the network. The RMs are responsible for calculating routes to information items and enforce cache allocation and replacement policies in the co-located node. Its decisions may be based on local information i.e. by monitoring the PIT table to estimate local information item demand, cache hit ratios, replacements or relevant information received by adjacent nodes for acquiring a network-wide view and take co-operative decisions with the other managers. A similar architecture with relevant distributed cache management algorithms have been presented in [18] while the use of traditional link state routing protocol has been described in [5]. In NDN nodes, the routing decisions are enforced by configuring the Forwarding Information Base (FIB) table of the NDN node so that interests follow the paths that the manager has calculated while the cache managers configures the Content Store (CS) management interface with the appropriate allocation, partitioning and replacement directives.

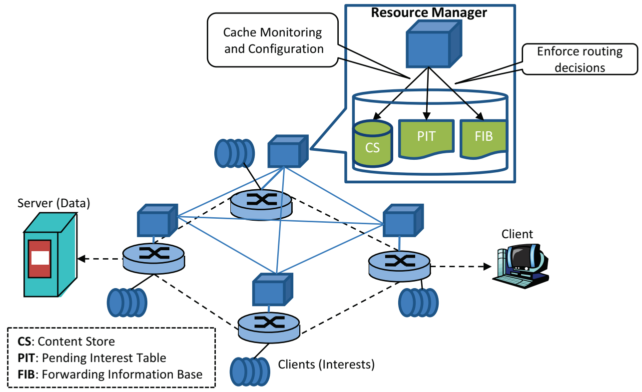


Fig. 2. ICN resource management in a NDN-based network architecture.

In the next section, we describe a routing algorithm appropriate for ICNs that takes into account cache related information and realizes the logic of the Topology and Resource Managers as described in the above system deployments. The routing decisions aim at minimizing an overall network-wide utility function i.e. the total transportation cost for the delivery of an item from the hosting server to a local client. This approach also assumes that every manager (in case of the distributed approach) has a network-wide view of the request patterns of every other node in the network and the cache hit ratio of each item at each node of the network.

## III. Cache Aware Routing Problem Formulation

We consider a network of arbitrary topology, represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ the communication links interconnecting them. Throughout the paper we will use the calligraphic letters to denote sets and the corresponding capitals for cardinality; for example $|\mathcal{V}| = V$. We also denote with $\mathcal{M}$ the set of the $M$ information items available at the network. For simplicity we assume that each item is of unit size.

Requests for content access are generated by the users of the network, with each user being directly connected to a node. Each information items $m$ is stored permanently at a node of the network (server $S^m$, $S \in \mathcal{V}$) and every request for that item from every other node is headed towards that node.

Every node $v \in \mathcal{V}$ has a cache of size $C_v$ slots (each slot can hold an item) and requests are generated by clients attached to the node with rate $r_v = \{r_v^1, \ldots, r_v^M\}$, where $r_v^m$ denotes the aggregate incoming request rate (in requests per second) at cache $v$ for information item $m$. Each node opportunistically caches passing by items following a given caching scheme and serves requests for items that are cached. Each item $m \in \mathcal{M}$ is cached with probability $p_v^m$ (cache hit ratio) at each node $v \in \mathcal{V}$, independently from node to node.

Assume that node $v_1$ needs to access the information item $m$ and chooses the path $p = (v_1, v_2, \ldots, v_n, v_{n+1})$ ($v_{n+1}$ is the server of item $m$, $p_{v_{n+1}}^m = p_{S^m} = 1$) from $v_1$ to $v_{n+1}$. We will use the notation $v_1 \rightsquigarrow_p v_{n+1}$ to represent a path from node $v_1$ to node $v_{n+1}$ (generally every path will be denoted by $\rightsquigarrow_p$).

The average number of hops to reach the item $m$ through path $p$ is:

$$\begin{aligned} H^m(v_1 \rightsquigarrow_p \; v_{n+1}) = 0 \cdot p_{v_1}^m + 1 \cdot p_{v_2}^m \cdot (1 - p_{v_1}^m) + \\ 2 \cdot p_{v_3}^m \cdot (1 - p_{v_2}^m) \cdot (1 - p_{v_1}^m) + \cdots + n \cdot p_{v_{n+1}}^m \cdot \\ (1 - p_{v_n}^m) \cdot \ldots \cdot (1 - p_{v_2}^m) \cdot (1 - p_{v_1}^m) \end{aligned} \tag{1}$$

where $n$ is the number of hops from node $v_1$ to node $v_{n+1}$ ($n+1$ is the number of nodes along the path).

For the $i$-th node in the path $p$, we define:

$$w_{\rightsquigarrow_p}^m(v_i) = \begin{cases} 0 & i = 1, \\ \prod_{j=1}^{i-1} (1 - p_{v_i}^m) & 2 \leq i \leq n+1. \end{cases} \tag{2}$$

We call $w_{\rightsquigarrow_p}^m(v_i)$ the *multiplicative weight* of node $v_i$ along path $p$ for the information item $m$ and $w_{\rightsquigarrow_p} = \prod_{j=1}^{n}(1 - p_{v_i}^m)$ the *multiplicative weight* along path $p$. The multiplicative weight of the whole path denotes also the probability of a request to reach the server of the requested item. Note that the multiplicative weight of a node $v_i$ depends only on the cache hit ratios of the items at the nodes in the path up to that node and not on the cache hit ratio of the node itself. Eq. 1 is transformed as:

$$\begin{aligned} H^m(v_1 \rightsquigarrow_p \; v_{n+1}) = H_{\rightsquigarrow_p}^m(v_{n+1}) = 0 \cdot p_{v_1}^m \cdot w_{\rightsquigarrow_p}^m(v_1) + \\ 1 \cdot p_{v_2}^m \cdot w_{\rightsquigarrow_p}^m(v_2) + 2 \cdot p_{v_3}^m \cdot w_{\rightsquigarrow_p}^m(v_3) + \cdots + \\ n \cdot p_{v_{n+1}}^m \cdot w_{\rightsquigarrow_p}^m(v_{n+1}) \end{aligned} \tag{3}$$

From the above equation is obvious that the average number of hops that a request will travel to reach the $i$-th node along the path $p$ is:

$$H_{\rightsquigarrow_p}^m(v_i) = \sum_{j=1}^{i} (j-1) \cdot p_{v_i}^m \cdot w_{\rightsquigarrow_p}^m(v_i) \quad 1 \leq i \leq n+1 \tag{4}$$

We also call the $H_{\rightsquigarrow_p}^m(v_i)$ the *transportation cost* of the unit size item $m$ from node $v_1$, where the request was made, to node $v_i$ along path $p$. Finally we denote as $H_{\rightsquigarrow_p}^m$ the *transportation cost* of the item $m$ along the path $p$.

We define the cheapest transportation cost of an item $m \in \mathcal{M}$ from node $v$, where the request was generated, to node $S^m$ (the server node for the particular item) as follows:

$$\delta(v, S^m) = \min\{H^m(v \rightsquigarrow_p S^m)\} = \min\{H_{\rightsquigarrow_p}^m\} \tag{5}$$

We assume that the graph $\mathcal{G}$ is connected and there is always a path from node $v \in \mathcal{V}$ to node $S^m$, $m \in \mathcal{M}$. The statement $\min\{H_{\rightsquigarrow_p}^m\}$ means that a path $p$ with the minimum transportation cost is selected among every path from node $v$ to node $S^m$. A cheapest path from $v$ to $S^m$ is defined as a path $p$ with $H_{\rightsquigarrow_p}^m = \delta(v, S^m)$. The cache aware routing problem is then defined as a problem to find the path with the minimum transportation cost for every unit size item $m \in \mathcal{M}$ from each node $v \in \mathcal{V} \backslash \{S^m\}$ to the hosting server $S^m$.

### A. Dynamic Programming approach

In the traditional shortest path problem settings the weight of each link is static and set to one. In the above setting the transportation cost between two neighboring nodes depends on their history on the path. From the Eq. 2-4 the transportation cost of item $m$ from node $v_i$ to node $v_{i+1}$ $((v_i, v_{i+1}) \in \mathcal{E})$ is $i \cdot \left(\prod_{j=1}^{i}(1-p_{v_i}^m)\right) \cdot p_{v_{i+1}}^m$. The multiplicative term $i \cdot \prod_{j=1}^{i}(1-p_{v_i}^m)$ is the dependence of the transportation cost between the two nodes to the history of the path.

From the formulation of the problem described above is obvious that:

$$H_{\rightsquigarrow_p}^m(v_{i+1}) = H_{\rightsquigarrow_p}^m(v_i) + i \cdot \left(\prod_{j=1}^{i}(1-p_{v_i}^m)\right) \cdot p_{v_{i+1}}^m \quad (6)$$

and for the minimum transportation cost of an item $m \in \mathcal{M}$ from node $v_1$, where the request was generated, to node $v_{i+1}$ (the server node for the particular item) we have:

$$\begin{aligned}
\delta(v_1, v_{i+1}) &= \min\left\{ H_{\rightsquigarrow_p}^m(v_i) + i \cdot \left(\prod_{j=1}^{i}(1-p_{v_i}^m)\right) \cdot p_{v_{i+1}}^m \right\} \\
&= \delta(v_1, v_i) + \min\left\{ i \cdot \left(\prod_{j=1}^{i}(1-p_{v_i}^m)\right) \cdot p_{v_{i+1}}^m \right\} \\
&= \delta(v_1, v_i) + i \cdot \min\left\{ \left(\prod_{j=1}^{i}(1-p_{v_i}^m)\right) \right\} \cdot p_{v_{i+1}}^m \\
&= \delta(v_1, v_i) + i \cdot \min\left\{ w_{\rightsquigarrow_p}^m(v_{i+1}) \right\} \cdot p_{v_{i+1}}^m \quad (7)
\end{aligned}$$

In order to minimize the multiplicative weight of a path, we choose as the next hop the node with the maximum cache hit probability, since this will minimizes the probability of a request to further be forwarded in the network (a item will be found in a intermediate node). The recursive application of Eq. 7 for each item $m \in \mathcal{M}$ and for each node $v \in \mathcal{V} \backslash \{S^m\}$ provides the paths with the minimum transportation cost for every item $m$ from every node $v$, where requests for that particular item were made, to the server $S^m$ of that item. At each iteration the minimum transportation cost towards a neighbor closer to the server is selected.

### B. Discussion

The DP algorithm presented above assumed that the cache hit ratio of each item at each node is independent from node to node. The cache hit ratio of each item at a given node depends on the caching scheme that is used. The caching scheme consists of two phases; in the first phase it is decided which passing by items to cache and in the second phase it is decided which cached item should be removed from the cache in case of an overflow. In order to have the required independency the caching scheme at each node should be based only on the local demand traffic and not in the passing by traffic. This means that the cache-everything-everywhere scheme presented in [5] could not be used with the DP algorithm, since the cache hit ratio of each item at each node strongly depends on the passing by traffic. In [19] authors consider the well known *Least Recently Used* (LRU) replacement algorithm and derive a closed-form formula that can be used for obtaining approximate hit/caching probabilities in constant time, i.e., without numeric computation that depends on input parameters like the number of objects and the storage capacity. In the approach presented in [19] the cache hit ratio of each item at each node is independent of the routing scheme and depends only on the local demand for each item and thus is used as the caching scheme for this work. We left for future work the adaptation of the DP algorithm to caching schemes similar to that presented in [5].

## IV. Performance Evaluation

In this section, we evaluate through simulations the performance of the proposed cache aware routing scheme (*CAWR*) and we compare it against the traditional shortest path routing scheme (*SHPT*) when the nodes use the LRU caching scheme and the cache hit ratio of each item at each node is given by the closed-form formula presented in [19]. Without loss of generality we assume that all caches have the same caching capacity ($C_v = C$, $\forall v \in \mathcal{V}$).

Each point of the following figures is the mean value out of fifty executions for fifty different network topologies. In particular, we use the Waxman model [20] to generate the network topologies, where a pair of nodes $\{u, v\}$ is connected with probability $p_{uv} = \beta e^{\frac{-d_{uv}}{L\alpha}}$ depending on their distance $d_{uv}$, with $L$ being the maximum distance of any two nodes and $\alpha, \beta \in (0, 1]$. Parameter $\beta$ controls the density of the graph (the larger the value of $\beta$ the denser is the graph), while $\alpha$ controls the connectivity of the graph (the smaller the value of $\alpha$ the larger is the number of short edges) [21]. Our fifty random network topologies arise from random values of parameters $\alpha$ and $\beta$.

Information-centric research lacks publicly available data sets for meaningful evaluation. Thus, synthetic workload generation according to realistic assumptions is widely accepted. Each information item at a given node is characterized by its *popularity*. Popularity refers to the request rate related to an item at the given node. We denote by $\vartheta_v$ the popularity of the items at node $v$. Popularity values are computed according to a Zipf law of exponents $z_{pop}$. Literature provides ample evidence that the file popularity in the Internet follows such a distribution [22]-[23]. Particularly, we assumed seven different parameters for the popularity exponents of the Zipf distribution ranging from $-1$ to $1$ ($\Theta = (-1, -0.75, -0.5, 0, 0.5, 0.75, 1)$). A Zipf distribution of negative exponent (e.g. $z_{pop} = -1$) means that out of the $M$ items in the network the most popular item in the $M$-th while the second most popular is the $(M-1)$-th and so on, with the first item being the least popular. On the other hand a Zipf distribution of positive exponent (e.g. $z_{pop} = 1$) means that the first item is the most popular and the $M$-th is the least popular. $z_{pop} = 0$ means that the items are of uniform popularity. The request rate of each item at each node varies from 0 - 250 requests per second according to the popularity of the item at the specific node. The seven Zipf distributions of popularities are assigned to the nodes of the network using a *locality* parameter $\lambda_\vartheta$, where $\vartheta \in \Theta$. Locality is also computed according to a Zipf law of exponent $z_{loc}$. The locality refers to the region of the topology where the seven different popularities are likely to originate. In particular,

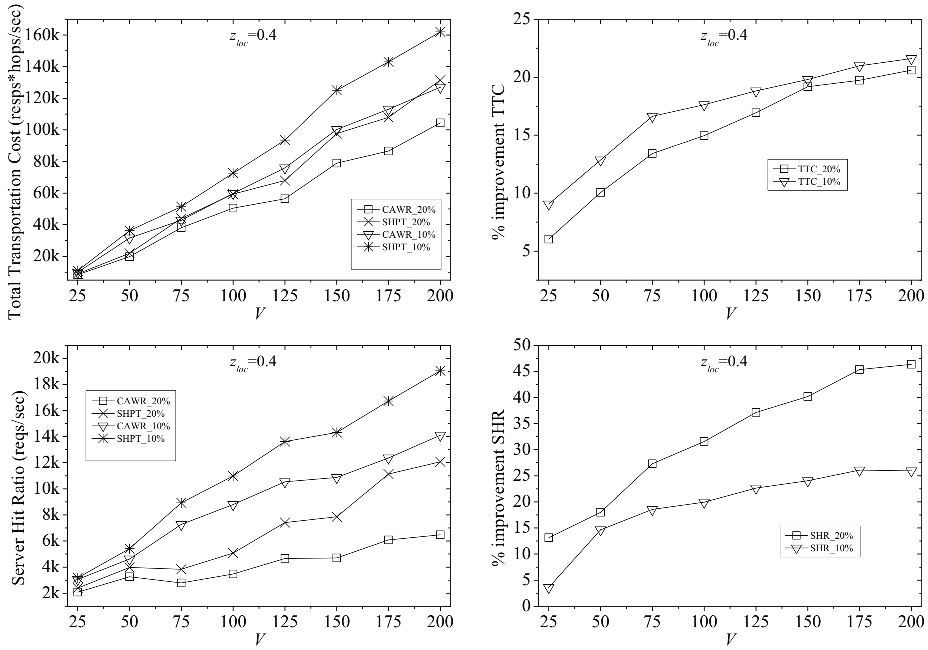


Fig. 3. The performance of the proposed cache aware routing scheme and the traditional shortest path routing vs. the number $V$ of nodes (caches) in the network, for two different capacities of the caches ($C/M = 20\%$ and $C/M = 10\%$).

$\lceil \lambda_\vartheta \cdot V \rceil$ nodes follow the popularity of exponent $\vartheta$. This set of nodes is computed by choosing randomly a central node and its $\lceil \lambda_\vartheta \cdot V \rceil - 1$ closest nodes, by executing a Breadth First Search algorithm as long as those nodes are not already assigned to a different popularity.

We are interested in the following performance metrics:

- The *Total Transportation Cost, TTC* (in $resps \cdot hops/sec$) for each one of the routing schemes for the transportation of all the items to the nodes, where the requests were made. This metric is the mean number of hops that all the requests (and the corresponding responses) will travel in the network until they reach the server or a cache where the requested item is cached.
- The *Server Hit Ratio, SHR* (in $reqs/sec$) for each one of the routing schemes. This metric is the mean number of requests that will finally reach the hosting server of the requested item, since the item could not be found at an intermediate cache.

The cache hit ratio of each item at each node depends only on the local demand and is independent form node to node, so the DP algorithm presented above could be applied for the computation of the cheapest paths.

In Fig. 3 we depict the impact of the number of nodes/caches $V$ in the network for two different capacities of the caches. The capacity of each cache is expressed as the fraction of the items that can be stored in the particular cache ($C_v/M = C/M = 10\%$ and $C/M = 20\%$). We notice that both the performance metrics exhibit a linear behavior of slope $V$. Also, the proposed CAWR scheme outperforms the traditional SHPT regardless of the size of the network or the caching capabilities of each cache. Particularly we observe $6\% - 18\%$ ($9\% - 19\%$ when $C/M = 20\%$) improvement regarding the total transportation cost even for very small networks, where the availability of alternative paths is small, and $5\%-20\%$ ($12\%-45\%$ accordingly) improvement regarding the Server Hit Ratio. In the case of the SHR when $C/M = 20\%$ the results are very impressive, where up to $45\%$ less requests, compared to the SHPT, finally reach the server implying a better utilization of the network resources and minimizing the needs for the deployment of server replicas in the network.

In Fig. 4 we depict the impact of cache capacity, expressed as the fraction of the items that can be stored in a cache for two different values of the locality exponent ($z_{loc} = 0.4$ and $z_{loc} = 0$). In general, we notice similar behavior for both metrics, with the proposed CAWR scheme performing better than the SHPT scheme. We also observe that the size of the neighborhoods, where the popularities are assigned, has not such a significant impact in the performance of the routing schemes with the the proposed CAWR performing slightly better when the neighborhoods are not of uniform size. Finally, in almost every experiment we observed that the SHR improvement is almost twice as much as the improvement in the TTC. This means that even if there are cases where the new routing scheme cannot alleviate the transportation costs, it can at least achieve significantly better utilization of the network resources and reduce the load at the hosting servers.

## V. Conclusion

In this paper, we proposed a new cache aware routing scheme for the computation of the paths with the minimum transportation cost in ICN, where in-network opportunistic caching is enabled. We particularly presented a Dynamic programming approach for the computation of the paths when

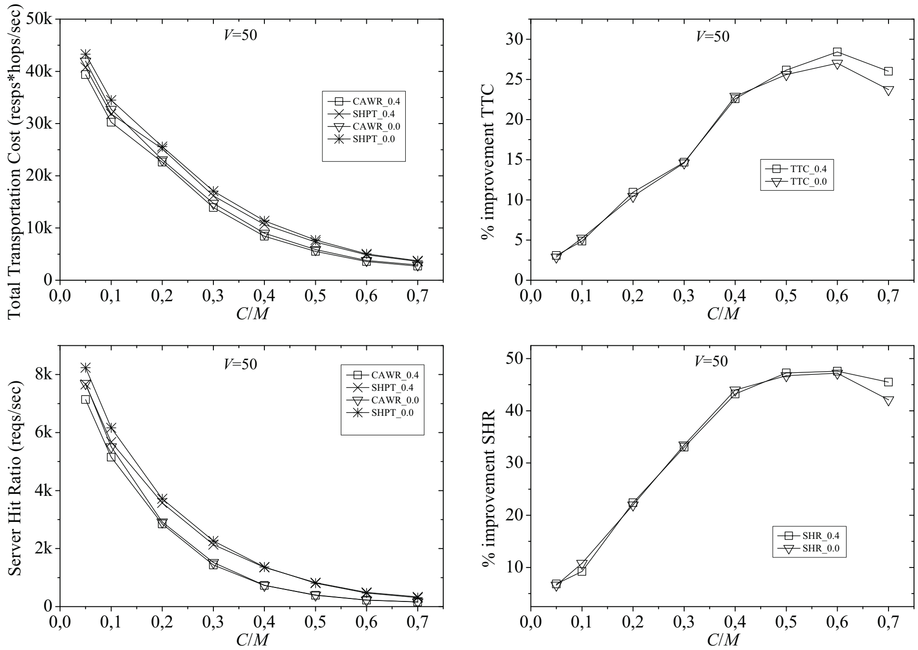


Fig. 4. The performance of the proposed cache aware routing scheme and the traditional shortest path routing vs. the fraction of the items that can be stored in a cache, for two different values of the locality exponent ($z_{loc} = 0.4$ and $z_{loc} = 0$).

the cache hit ratio of each item at each node change based on the observed item request patterns such as their popularity and locality and the used caching scheme. Finally, we present a resource management architecture for the cache aware routing in an ICN, where distributed resource managers make route decisions, and we validate the proposed approach through simulations.

It is evident that the use of alternative routes other than the shortest path (in terms of hops) with different caching capabilities for each item give significant performance benefits and reduce significantly the total transportation cost and the load of the hosting server. It would be interesting, as future work, to explore enhancements to the proposed routing scheme that would also take into consideration the presence of server replicas.

## ACKNOWLEDGMENT

V. Sourlas' work is co-financed by the European Union (European Social Fund ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning of the National Strategic Reference Framework (NSRF) Research Funding Program: Heracleitus II - Investing in knowledge society through the European Social Fund.

## REFERENCES

[1] Named Data Networking project, available at http://named-data.net, 2011.

[2] PURSUIT project, available at http://www.fp7-pursuit.eu, 2011.

[3] W. K. Chai, N. Wang, I. Psaras, G. Pavlou, Ch. Wang, G. Garca de Blas,F. Salguero, L. Liang, S. Spirou, An. Beben and El. Hadjioannou, "CURLING: Content-ubiquitous resolution and delivery infrastructure for next-generation services," IEEE Commun. Mag., vol. 49, no. 3, pp. 112-120, 2011.

[4] B. Ohlman et al, "First NetInf architecture description," Apr. 2009. http://www.4ward-project.eu/index.php?s=file_download&id=39.

[5] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. Briggs, R. Braynard, "Networking named content," Proceedings of the 5th ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT 2009), Rome, Italy, Dec. 1-4, 2009.

[6] L. Muscariello, G. Carofiglio, and M. Gallo, "Bandwidth and storage sharing performance in information centric networking," in ACM SIGCOMM ICN Workshop, 2011, pp. 26–31.

[7] U. Lee, I. Rimac, and V. Hilt, "Greening the internet with content-centric networking," eEnergy, p. 179, 2010.

[8] E. J. Rosensweig and J. Kurose, "Breadcrumbs: Efficient, Best-Effort Content Location in Cache Networks," in INFOCOM 2009.

[9] D. Perino and M. Varvello, "A reality check for content centric networking," in ACM SIGCOMM ICN Workshop, 2011, pp. 44–49.

[10] S. Arianfar, P. Nikander, and J. Ott, "On content-centric router design and implications," in ReArch Workshop, vol. 9. ACM, 2010, p. 5.

[11] Al. Ghodsi, T. Koponen, B. Raghavan, S. Shenker, An. Singla and J. Wilcox, "Information-centric Networking: Seeing the forest for the trees", ACM Workshop on Hot Topics in Networks (HotNets-X), Cambridge, MA, Nov. 2011.

[12] W. K. Chai, I. Psaras and G. Pavlou, "Cache Less for More In Information-Centric Networks," IFIP NETWORKING 2012, Prague, Czech Republic, May 2012.

[13] S. Dibenedetto, C. Papadopoulos and D. Massey, "Routing Policies in Named Data Networking," in proc. of ACM SIGCOMM workshop on Information-centric networking, pp. 38-43, 2011.

[14] H. Liu, X. De Foy and D. Zhang, "A Multi-Level DHT Routing Framework with Aggregation," in proc. of ACM SIGCOMM workshop on Information-centric networking, 2012.

[15] S. Eum, K. Nakauchi, M. Murata, Y. Shoji and N. Nishinaga, "CATT: Potential Based Routing with Content Caching for ICN," in proc. of ACM SIGCOMM workshop on Information-centric networking, 2012.

[16] R. G. C. Intanagonwiwat and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in proc. of Mobile computing and networking, Boston, USA, Aug 2000.

[17] V. Lenders, M. May and B. Plattner, "Service discovery in mobile ad hoc networks: A field theoretic approach," Pervasive and Mobile Computing, vol. 3, pp. 343–370, 2005.

[18] V. Sourlas, P. Flegkas, L. Gkatzikis and L. Tassiulas, "Autonomic Cache Management in Information-Centric Networks," in 13th IEEE/IFIP Network Operations and Management Symposium (NOMS 2012), pp. 121–129, Hawaii, USA, April 2012.

[19] N. Laoutaris, "A Closed-Form Method for LRU Replacement under Generalized Power-Law Demand," arXiv:0705.1970v1, 2007.

[20] B. Waxman, "Routing of multipoint connections," IEEE JSAC vol. 6, pp. 1617–1622, 1998.

[21] P. Habegger, H. Bieri, "Modeling the topology of the Internet: an assessment," Technical Report, Institut fr Informatik und angewandte Mathematik, University of Bern, Switzerland, IM-02-2002, 2002.

[22] L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker, "Web caching and Zipf-like distributions: evidence and implications," In IEEE INFOCOM, NY, March 1999.

[23] L. A. Adamic and B. A. Huberman, "Zipfs law and the Internet," Glottometrics, vol. 3, pp. 143–150, 2002.