Optimal Design of Information Centric Networks

Michele Mangili[a,b], Fabio Martignon[a,c], Antonio Capone[b,*]

[a] LRI, Université Paris-Sud, Bat. 650, rue Noetzlin, 91405 Orsay, France.
[b] DEIB, Politecnico di Milano, via Ponzio 34/5, 20133 Milano, Italy.
[c] IUF, Institut Universitaire de France

Abstract

Information-Centric Networking (ICN) has recently gained momentum as a promising paradigm for the next-generation Internet architecture. The first prototypes for ICN-capable routers have already been developed; however, to migrate the devices to this novel architecture, non-negligible investments should be made. Therefore, it is of utter importance to provide clear quantitative insights of the expected economic benefits that operators will experience by switching to the ICN paradigm. For these reasons, in this paper we tackle the *content-aware network-planning* problem, and we formulate a novel optimization model to study the migration to an ICN, in a budget-constrained scenario. Our formulation takes into accurate account traffic routing and content caching. We prove that the optimization problem is NP-Hard, then we formulate heuristics to efficiently solve it. An extensive simulation campaign with real network topologies shows that our greedy heuristic cuts the computation time while finding close to optimal solutions, and therefore can effectively support network operators to evaluate the effects of a migration to ICN.

Keywords: Information-Centric Networking, Planning, Optimization, Content Distribution.

*Corresponding author, Tel: (+39) 02 2399 3449, Fax: (+39) 02 2399 3413
Email addresses: michele.mangili@lri.fr (Michele Mangili), fabio.martignon@lri.fr (Fabio Martignon), capone@elet.polimi.it (Antonio Capone)

1. Introduction

Recent traffic measurements have clearly shown that more than 50% of the overall Internet traffic is generated to retrieve contents, as illustrated in the Sandvine Report and Cisco VNI forecasts [1, 2]. However, being able to accommodate the content distribution needs of the users is still in today's Internet a challenging task, and adequate technical solutions such as Content Delivery Networks (CDNs) have specifically been designed to achieve this objective [3, 4].

Meanwhile, innovative paradigms known under the name of Information-Centric Networking (ICN) have recently gained momentum in the research community. Despite the fact that there are many different designs that belong to this category, all of them are based on the idea that by directly intervening on the protocol stack, the content distribution capabilities of the network can be boosted [5]. Among all the advantages that can be experienced by switching to this novel infrastructure, *traffic offloading* stems out as being the most relevant achievement [6]. Despite that, other advantages can also be gained: lower delays, better security and multipath routing all integrate as positive features of these networks [7, 8].

Rather than being in their preliminary steps, these research efforts have already reached the point where the first working prototypes for ICN-enabled routers have been realized by Alcatel [9], Cisco [10] and Parc [11]. Specific hardware and software components are required in order to support ICN packet forwarding at wire-speed, and thus operators will certainly have to make non-negligible investments to purchase the new ICN devices. As a result, they will be willing to transition their infrastructures to ICN only if clear economic benefits are envisioned: by upgrading a router to ICN and by installing a given amount of storage to memorize frequently requested contents, it will directly serve incoming requests for the cached objects. In this way, given the fact that the content popularity is very skewed (i.e: few objects generate most of the traffic [12, 13]), the operator can experience significant economic savings accountable to traffic offloading [14]. While CDNs may also be used to efficiently serve clients' requests, they are usually regarded as an expensive solution, since they demand to centrally orchestrate replica placement and request routing [15], while in ICN each network device will autonomously perform these choices, thus reducing the overall management costs.

To pave the way for a potential paradigm shift from a TCP/IP network to ICN, we specifically consider the *migration* step to the ICN architecture and we formulate a novel *content-aware network planning model* that we use to compute the optimal *migration strategy* for the operator. On top of that, by considering relevant economic parameters, our model can also be used to understand which economic benefits are expected as a result of the transition to ICN. To achieve all these objectives, we take into account three economic parameters: (1) a traffic-proportional link cost, (2) the router migration cost and (3) the storage cost, proportional to the amount of memory installed at a given ICN-migrated node.

To summarize, in this paper we provide the following contributions:

1. We formulate a model to evaluate the optimal content-distribution performance of an IP network under unsplittable routing conditions.

2. We propose a novel *content-aware network-planning* Mixed Integer Linear Programming (MILP) model for the *migration* to an ICN. Our formulation determines the optimal node migration and cache allocation in a budget-constrained scenario. Unsplittable routing conditions are still enforced by non-migrated routers.

3. We prove that the content-aware network-planning problem is NP-Hard, therefore we propose a novel and very efficient *greedy* heuristic, that outperforms the *randomized rounding* algorithm we designed in [16].

4. We compare the performance of the *randomized rounding* heuristic with the new *greedy* solver, showing that the latter dramatically improves the quality of the final solution while cutting the computation time of the heuristic of at least an order of magnitude.

5. We quantitatively evaluate the benefits of migrating to ICN, with different budgets as well as pricing configurations.

Our key findings suggest that (1) for a very large span of pricing policies, by migrating only few nodes to ICN remarkable traffic reductions will be experienced by the operator; (2) ICN benefits also content providers since it significantly offloads their distribution infrastructures, and (3) when the content popularity distribution is very skewed (i.e: most of the traffic is generated by few popular objects), the storage space installed at the migrated nodes is an order of magnitude smaller than for less skewed distributions.

This work highlights the importance of performing an economic analysis of the advantages that can be obtained by migrating to ICN, while performing a content-aware network-planning and explicitly taking into account the migration, storage and traffic costs.

This paper is structured as follows: in Sec. 2 we introduce the system model. In Sec. 3 we extensively describe the optimization models we use to compute the overall content delivery cost of an IP network and the content-aware planning model for the migration to an ICN. In Sec. 4 we illustrate our proposed *randomized rounding* and *greedy* heuristics for ICN, while numerical results are discussed in Sec. 5. Related works are presented in Sec. 6, and finally, concluding remarks are illustrated in Sec. 7.

2. System Model

In this section we describe the system model and discuss the rationale of our approach. A comprehensive introduction to some of the most notable ICN proposals can be found in [6].

Fig. 1 represents an example to describe relevant features of our proposed system model. Three types of nodes are available in the topology: consumers, producers and routers. All the nodes operate on a finite set of contents, called the *"catalog"*. For the sake of simplicity, as depicted in Fig. 1, in this example we assume that the catalog is composed of 5 objects. Producers publish objects

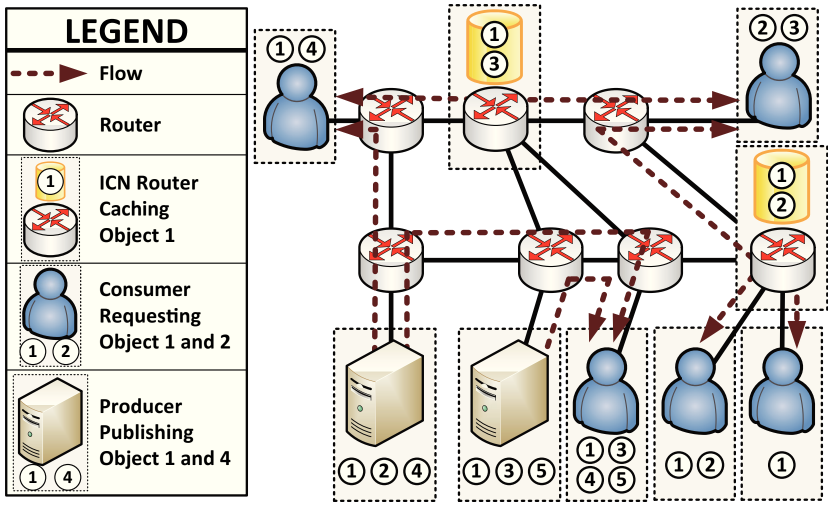


Figure 1: *System model.* Given the network topology, consumers' requests, and objects served by content providers, our optimization model chooses which routers should be migrated to ICN, and which objects should they cache.

in the network, whereas consumers generate demands for them. It is possible that the same object is provided by different producers, as represented in the figure.

Each link in the network is characterized by having a traffic-proportional price (OPEX) and a bounded capacity. Since ICN routers can provide in-network caching functionalities, the operator can significantly reduce its traffic costs, by migrating routers to this paradigm and by moving content replicas closer to the location where most of the users are requesting them. However, to perform the migration, the operator must pay the corresponding costs (CAPEX) which are given by (1) the price to migrate a router to ICN, C^M and (2) the storage price to memorize one object at a migrated router, C^S. We bound the migration costs (i.e, those due to node migration and caching storage) to the value B, that is the total migration budget the operator is willing to spend. ICN routers issue upstream traffic requests as if they were the request origins; finally, on top of offering caching functionalities, they also support splittable request routing at the level of the single object requests. We assume that any router in the topology can be migrated to ICN, however it is very simple to extend our models and algorithms to restrict the set of routers eligible for the migration.

Intra-domain routing protocols such as OSPF provide equal-cost multipath (ECMP) functionalities, to support flow splitting over multiple paths having equal-cost weights [17]. However, in this work we formulate the optimization problem for an IP network by explicitly considering unsplittable routing conditions. There are many reasons behind this choice: first of all, to the best of our knowledge, network operators are often still reluctant to actually take advantage of multipath functionalities, since having to deal with single-path flows facilitates network management and troubleshooting [18, 19, 20]. Secondly, there are some scenarios in which avoiding packet reordering becomes a major requirement and therefore multipath flow splitting should be prevented [19]. On top of that, the focus of our contribution is to consider the overall content distribution costs expressed as a function of the link costs which are provided as input parameters and are fixed; on the other hand, in order to take advantage of ECMP functionalities, the link weights are dynamically changed by the operator as a function of the actual congestion. However, for the sake of completeness, in Sec. 5.6 we also take into account the case in which unsplittable traffic conditions are relaxed, showing that, on average, their impact on the overall cost is rather limited.

In this work we are interested in studying the performance bounds that can be achieved by migrating a subset of the available routers to ICN. In order to do that, we consider the *off-path* caching scenario, as shown in Fig. 2.

In particular, while in *on-path* caching flows are forwarded on the shortest path to the closest producer publishing the requested content, in *off-path* caching network nodes have a "full" visibility of the contents stored in each ICN router. Therefore, a node can eventually divert traffic requests on a path where a copy of the content can be retrieved, saving the cost to contact the original producer.

We assume that the cost to move content replicas to the intermediate nodes is very small compared to the one that network operators need to sustain on a daily basis to serve consumers' demands. As a matter of fact, in this work

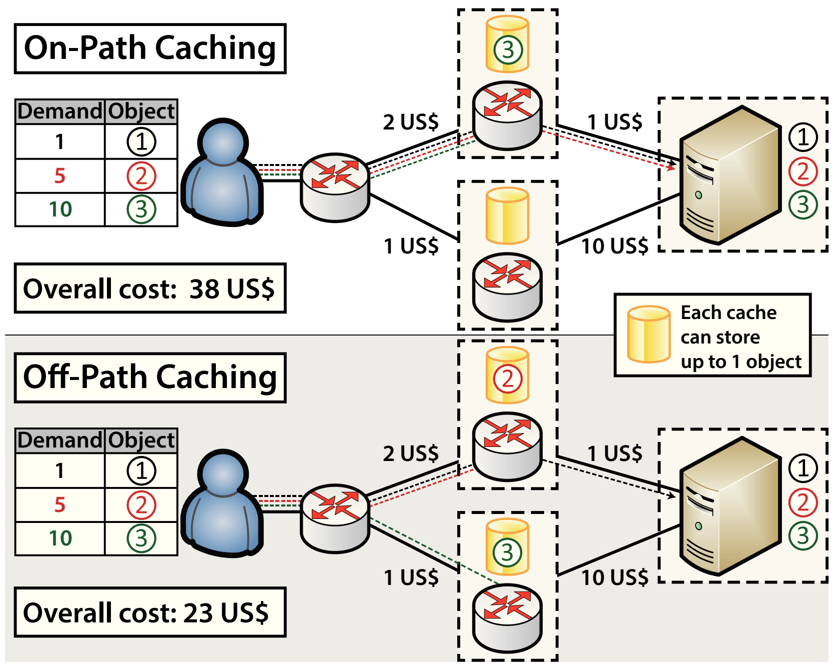


Figure 2: *On-path vs. Off-path caching.* In the example, the upper path is the shortest to the producer (in terms of cost), moreover each router can store up to 1 object in its cache. The overall cost is the sum of the product of the traffic demand and the link cost on which flows are forwarded. In the *on-path approach* only the shortest path can be used to serve the demands and the minimum cost solution of 38 US$ is obtained by storing object *3* in the intermediate router. On the other hand, in the *off-path approach*, also the lower path can be used and the minimum cost solution improves to the lower value of 23 US$, that is achieved by placing object *3* and *2* in the caches of the lower and upper router, respectively.

we consider the long-term planning problem of the networking infrastructure, and we assume that throughout the entire life-span considered cached contents will not be refreshed. Therefore, the cost to initially push the contents to the intended caches is amortized over time and becomes negligible.

In Fig. 2 a toy example showing this positive advantage is represented. In fact, while in on-path caching the overall cost[1] to serve traffic demands is 38 US$,

[1] As detailed in Sec. 3, the overall cost is given by the sum of the product of the forwarded traffic and the link cost.

in off-path caching only 23 US$ are required. On the other hand, the computational complexity of the object-placement problem is significantly increased due to the fact that object placement and routing must be jointly optimized network-wide.

In the rest of the paper we describe our proposed optimization models and heuristics to help the operator determine the optimal ICN node migration strategy, object placement and request routing, considering off-path caching. We focus on this case since it is the one that leads to the best performance bounds, even though our models and algorithms can easily be extended to the on-path caching scenario.

3. Design Models

We now describe the optimization models we use to evaluate the migration to an ICN. Sec. 3.1 presents the IP network model, while Sec. 3.2 is devoted to the ICN network planning formulation, as well as the formal proof that the two problems are NP-Hard.

Let us introduce the notation used in describing the planning problems and in the optimization models. We represent the network as a directed graph $G = (N, A)$, where the set of nodes N is partitioned into consumers C, producers P, and routers R (i.e, $N = C \cup P \cup R$).

The set of forward and backward arcs of node $i \in N$ are denoted with $FS(i)$ and $BS(i)$, respectively. Network arcs $(i, j) \in A$ are characterized by a capacity, denoted with $b_{i,j}$, and a price per unit of traffic, $p_{i,j}$. This is representative of the prices charged by services like Amazon CloudFront [21], as we discuss in Sec. 5.1. We denote with O the set of objects, and we assume that all of them have the same size, as frequently done in the literature (e.g: [22, 23]). Let Q be the set of requesters; for both the IP and ICN network models, requesters are nodes from which traffic requests originate: in the IP network, only the consumers can behave as such, and thus $Q \equiv C$. Each consumer $c \in C$ expresses a traffic demand d_c^o for object $o \in O$. Producers can serve a subset of the entire object

Table 1: Summary of the notation used in this paper.

Parameters of the Models	
A	Set of arcs
$N, C,$ P, R	N Set of nodes, $C \subset N$ Set of consumers, $P \subset N$ Set of producers, $R \subset N$ Set of routers
Q	Set of requesters. In the IP network model $Q = C$. In the ICN model $Q = C \cup R$.
O	Set of objects
$FS(i)$	Set of forward arcs $(i, j) \in A$ for node $i \in N$
$BS(i)$	Set of backward arcs $(j, i) \in A$ for node $i \in N$
$b_{i,j}$	Capacity of arc $(i, j) \in A$
$p_{i,j}$	Price per unit of traffic on arc $(i, j) \in A$
d_c^o	Demand of consumer $c \in C$ for object $o \in O$
r_p^o	0-1 Object reachability matrix $r_p^o = 1$ if producer $p \in P$ can serve object $o \in O$
C^M	Price to migrate one router to ICN
C^S	Price to install one unit of storage
B	Total migration budget

Decision Variables of the Models	
$y_{i,j}^{o,q}$	Flow on arc $(i, j) \in A$ for object $o \in O$, requested by requester $q \in Q$
$z_{i,j}^q$	0-1 Routing variable: $z_{i,j}^q = 1$ if arc $(i, j) \in A$ can be used to route requests for requester $q \in Q$
m_r	0-1 Router migration variable $m_r = 1$ if router $r \in R$ migrates to ICN.
k_r^o	0-1 Cache storage variable $k_r^o = 1$ if ICN router $r \in R$ caches object $o \in O$
w_l^o	Flow served by producer or router $l \in (P \cup R)$ for object $o \in O$, when l stores a replica of o
$F_r^{o,q}$	Flow balance at router $r \in R$, for object $o \in O$, requested by $q \in Q$

catalog, in particular we represent with the binary parameter r_p^o the object reachability matrix ($r_p^o = 1$ if producer $p \in P$ publishes object $o \in O$, otherwise $r_p^o = 0$). For the sake of clarity, in Table 1, we summarize the notation used throughout the paper.

3.1. IP Network model

We start by describing the IP network model we use as a benchmark with respect to the solution we get when studying the planning of an ICN. In the IP routing problem, objects must be routed from producers where they are available to consumers, possibly passing through routers, at the minimum overall cost. We assume that flows are unsplittable.

The problem can be naturally described as a multicommodity flow model, where a commodity is associated with every pair {object, requester}. Let variables $y_{i,j}^{o,q}$ denote the flow of object $o \in O$ on arc $(i,j) \in A$ for requester $q \in Q$. In addition, in order to account for the unsplittable flow requirement, we introduce binary variables $z_{i,j}^q$ whose value is 1 if arc $(i,j) \in A$ is used to route traffic for requester $q \in Q$, and 0, otherwise. Note that another aspect of the problem involves the selection of the producer to serve each request, in the presence of multiple copies of some objects. To account for that, variables w_p^o denoting the actual quantity of flow of object o referring to producer p must be introduced.

The minimum cost request routing problem under unsplittable flow conditions for an IP network can therefore be formulated as follows:

$$\min \sum_{(i,j)\in A} p_{i,j} \sum_{o\in O} \sum_{q\in Q} y_{i,j}^{o,q} \tag{1}$$

subject to:

$$\sum_{(j,r)\in BS(r)} y_{j,r}^{o,q} - \sum_{(r,j)\in FS(r)} y_{r,j}^{o,q} = 0 \qquad \forall o \in O, \forall q \in Q, \forall r \in R \quad (2)$$

$$\sum_{(j,i)\in BS(i)} y_{j,i}^{o,i} = d_i^o \qquad \forall o \in O, \forall i \in C \quad (3)$$

$$\sum_{q\in Q} \sum_{(p,j)\in FS(p)} y_{p,j}^{o,q} = w_p^o \qquad \forall o \in O, \forall p \in P \quad (4)$$

$$w_p^o \leq r_p^o \sum_{c\in C} d_c^o \qquad \forall p \in P, \forall o \in O \quad (5)$$

$$\sum_{c\in C} d_c^o = \sum_{p\in P} w_p^o \qquad \forall o \in O \quad (6)$$

$$\sum_{o\in O} \sum_{q\in Q} y_{i,j}^{o,q} \leq b_{i,j} \qquad \forall (i,j) \in A \quad (7)$$

$$\sum_{o\in O} y_{i,j}^{o,q} \leq b_{i,j} z_{i,j}^q \qquad \forall i \in N \setminus C, \forall (i,j) \in FS(i), \forall q \in Q \quad (8)$$

$$\sum_{(i,n)\in FS(i)} z_{i,n}^q \leq 1 \qquad \forall i \in N \setminus C, \forall q \in Q \quad (9)$$

$$z_{i,j}^q \in \{0,1\} \qquad \forall q \in Q, \forall (i,j) \in A \quad (10)$$

$$w_p^o \geq 0 \qquad \forall p \in P, \forall o \in O \quad (11)$$

$$y_{i,j}^{o,q} \geq 0 \qquad \forall o \in O, \forall q \in Q, \forall (i,j) \in A. \quad (12)$$

The objective function (1) minimizes the overall traffic costs incurred by the provider on all network arcs.

The flow balance at every router and consumer node is imposed by (2) and (3), respectively. The balance at producer nodes depends on the requested flow of each object (4) which is regulated by (5) and (6). These constraints consider the fact that requests can be served only by those producers that are actually publishing a copy of the given object in the network, and that the overall traffic served by the producers equals the overall demand expressed by the consumers.

Capacity constraints are enforced in (7). Unsplittable routing conditions are imposed in (8) and (9). In particular, the set of constraints (8) makes sure that flows for requester $q \in Q$ are forwarded only on the arcs $(i,j) \in A$ where $z_{i,j}^q = 1$, whereas in (9) we make sure that routers and producers have at most only one egress arc for requester $q \in Q$.

Finally, non negativity on flow variables and binary condition on $z^q_{i,j}$ are imposed in (10)-(12). Notice that if 0-1 variables $z^q_{i,j}$ are fixed, the problem amounts to a continuous multicommodity flow that can be solved by standard linear programming solvers.

3.2. ICN Planning

We now extend the model presented in Sec. 3.1 to solve the *content-aware network planning* problem in ICN.

Let C^M denote the additional cost to migrate one IP router to ICN. Once a router has been migrated to this paradigm, caching storage can be installed on it. C^S denotes the storage cost to add the caching space sufficient to memorize one object. The overall *migration cost* should not exceed the total available budget, which is denoted with B. Two sets of binary variables are used in the ICN planning model: m_r and k^o_p. They are such that $m_r = 1$ if router $r \in R$ migrates to ICN, otherwise $m_r = 0$; similarly $k^o_r = 1$ if router $r \in R$ caches object $o \in O$, while $k^o_r = 0$ if the object is not cached.

With $F^{o,q}_r$ we denote the flow balance at router $r \in R$ for object $o \in O$ requested by $q \in Q$. If $F^{o,q}_r > 0$ then it means that $r \in R$ generates demands for object o and thus it is leveraging multipath routing functionalities; on the other hand, if $F^{o,q}_r < 0$ the node behaves as a source node, since it is caching object o and is serving traffic requests for it; finally if $F^{o,q}_r = 0$, r is only forwarding flows for o.

Given the above definitions we formulate the budget-constrained ICN planning problem (BC-ICN) as follows:

$$\min \sum_{(i,j)\in A} p_{i,j} \sum_{\substack{o\in O\\ q\in Q}} y^{o,q}_{i,j} + \left[C^M \sum_{r\in R} m_r + C^S \sum_{r\in R}\sum_{o\in O} k^o_r \right] \tag{13}$$

subject to (3)-(5), (7)-(8), (10), (12), and

$$\sum_{(j,r)\in BS(r)} y_{j,r}^{o,q} - \sum_{(r,j)\in FS(r)} y_{r,j}^{o,q} = F_r^{o,q} \qquad \forall o \in O, \forall q \in Q, \forall r \in R \tag{14}$$

$$-\sum_{q\in Q} F_r^{o,q} = w_r^o + \sum_{(i,r)\in BS(r)} y_{i,r}^{o,r} \qquad \forall o \in O, \forall r \in R \tag{15}$$

$$w_r^o \leq k_r^o \cdot \sum_{c\in C} d_c^o \qquad \forall o \in O, \forall r \in R \tag{16}$$

$$k_r^o \leq m_r \qquad \forall o \in O, \forall r \in R \tag{17}$$

$$y_{i,j}^{o,r} \leq m_r b_{i,j} \qquad \forall (i,j) \in A, \forall o \in O, \forall r \in R \tag{18}$$

$$\sum_{(i,j)\in FS(i)} z_{i,j}^q \leq 1 \qquad \forall i \in P, \forall q \in Q \tag{19}$$

$$\sum_{(r,j)\in FS(r)} z_{r,j}^q \leq 1 + m_r \cdot (|FS(r)| - 1) \qquad \forall r \in R, \forall q \in Q \tag{20}$$

$$\sum_{c\in C} d_c^o = \sum_{l\in P\cup R} w_l^o \qquad \forall o \in O \tag{21}$$

$$C^M \sum_{r\in R} m_r + C^S \sum_{r\in R}\sum_{o\in O} k_r^o \leq B \tag{22}$$

$$y_{r,i}^{o,r} = 0 \qquad \forall r \in R, \forall (r,i) \in FS(r), \forall o \in O \tag{23}$$

$$w_l^o \geq 0 \qquad \forall l \in P \cup R, \forall o \in O. \tag{24}$$

The objective function (13) takes into account traffic and migration cost components. The former is given by $\sum_{(i,j)\in A} p_{i,j} \sum_{\substack{o\in O\\ q\in Q}} y_{i,j}^{o,q}$, the latter is instead the sum of node migration costs $C^M \sum_{r\in R} m_r$, and storage costs $C^S \sum_{r\in R}\sum_{o\in O} k_r^o$.

Flow balance constraints for routers are expressed in (14). In particular, if a router $r \in R$ migrates to ICN (i.e, $m_r = 1$), we let the flow balance be $F_r^{o,q} \leq 0$, meaning that r can directly serve incoming requests; otherwise, if $m_r = 0$ we set $F_r^{o,q} = 0$. The set of constraints (15) permits a router $r \in R$ to have caching functionalities (i.e, $w_r^o \geq 0$); furthermore it lets r behave as a requester (i.e, $y_{i,r}^{o,r} \geq 0$), a feature that facilitates traffic splitting in the network. The joint presence of constraints (16)-(18) makes sure that only ICN-migrated routers can provide caching functionalities and behave as requesters. In-network caching features are modeled in (16) and (17). In particular, if a router r migrates to ICN and stores in its local cache a copy of object o, it is then capable of directly serving upstream requests for that particular object. Instead, in (18) we prevent non-migrated routers to behave as requesters.

Unsplittable request routing is enforced in the set of constraints (19) (*for producers only*) and (20) (*for routers only*). In particular, this latter set of constraints lets a migrated ICN router $r \in R$ support splittable routing: if the router does not migrate to ICN (i.e, $m_r = 0$), then at most one egress link is used to route requests for $q \in Q$, otherwise if $m_r = 1$, then all the egress links can be used, making the ICN-migrated router capable to perform multipath routing. In (21), we impose the condition that the overall demand generated for object $o \in O$ is satisfied by producers and caching routers. The budget allocated for the migration is limited by (22). The set of constraints (23) avoids loops, preventing requests expressed by a router to be fulfilled by the router itself, while in (24) non-negativity on flow variables is enforced.

We now study the computational complexity of our proposed BC-ICN problem and we formally prove that it is NP-Hard.

Theorem 1. *The budget-constrained ICN planning problem (BC-ICN) is NP-Hard.*

Proof. In the *single-source unsplittable flow problem* (UFP) we want to find a feasible unsplittable routing for all the commodities of a network $G = (V, E, u)$, given a source vertex $s \in V$, a set of k commodities with sinks $t_1, \ldots, t_k$ and a corresponding real-valued demand $\rho_1, \ldots, \rho_k$. The unsplittable routing condition is enforced by routing the ρ_i demand on a single $s - t_i$ flow path. The feasibility question for UFP is strongly NP-complete [24].

Consider any instance of UFP. We polynomial-time reduce it to BC-ICN as follows: first of all we set unitary link-prices, as well as $C^M = 1, C^S = 1, B = 0$. We then create a new network with one producer, corresponding to the single source vertex $s \in V$ for UFP, and k consumers, one for each of the commodities available. The object catalog is composed of the k commodities (i.e, $|O| = k$).

Consumers' demands are set as follows:

$$\begin{cases} d_c^o = \rho_c & \forall c \in C, o \in O \;\; o = c \\ d_c^o = 0 & \forall c \in C, o \in O \;\; o \neq c \end{cases} \tag{25}$$

Since we reduced UFP to BC-ICN in polynomial-time, BC-ICN is NP-Hard.

□

Furthermore, it is easy to show that UFP can be reduced in polynomial-time to the optimal planning problem of the IP network we presented in Sec. 3.1, therefore we conclude that both optimization problems are NP-Hard.

4. Heuristics

The network planning problems we introduced in the previous section are NP-Hard and, as we will show in Sec. 5, even by using best of breed ILP solvers available today, it is often very challenging, in terms of computation time, to find the optimal ICN planning solution (as we discuss in Sec. 5.3). Therefore, there is the clear need to formulate heuristics that can efficiently find a close to optimal solution for the network planning problem. While the computation time is not a concern per-se (since we are going to run the algorithms off-line), our main goal while formulating the heuristics is to be able to solve very large network instances. To this aim, in the following Sec. 4.1 we briefly describe the randomized rounding heuristic we originally formulated in our previous work [16]; then in Sec. 4.2 we illustrate our novel greedy heuristic that outperforms the randomized rounding algorithm, as we extensively show in the numerical comparison in Sec. 5.

4.1. Randomized Rounding Heuristic

Algorithm 1 illustrates the randomized rounding heuristic in pseudo-code. The rationale behind it is to solve the continuous relaxation of the ICN model described in Sec. 3.2, computing the optimal fractional values of $\widehat{k}_r^o$. We then interpret $\widehat{k}_r^o$ as the probability that object $o \in O$ is placed in the cache of the

Algorithm 1: Randomized Rounding

Input : $mdl \Leftarrow \langle b_{i,j}, p_{i,j}, d_c^o, r_p^o, C^M, C^S, B \rangle$

Output: obj_fun

1 $\widehat{k}_r^o \Leftarrow$ SolveRelaxedICNModel(mdl);

2 $\widehat{k}_r^{max} \Leftarrow \max_{\forall o \in O} \widehat{k}_r^o$;

3 $RL \Leftarrow$ SortRoutersByCumulativeProbabilityPerObject($\widehat{k}_r^o$);

$C \Leftarrow 0$;

foreach $r \in RL$ **do**

  $\bar{m}_r \Leftarrow false$;

  foreach $o \in O$ **do**

4    $w \Leftarrow \left\{ \text{UniformRndValue}(0,1) \leq (\widehat{k}_r^o / \widehat{k}_r^{max}) \right\}$;

5   **if** $w \wedge (C < B)$ **then**

6   **if** $\neg\, \bar{m}_r$ **then** $C \Leftarrow C + C^M$;

7      $C \Leftarrow C + C^S; \bar{k}_r^o \Leftarrow true; \bar{m}_r \Leftarrow true$;

    end

  end

end

8 $obj\_fun \Leftarrow$ SolveICNModel($mdl, \bar{k}_r^o, \bar{m}_r$);

migrated router $r \in R$. As frequently done in the randomized rounding literature [25], we scale the relaxed variables for object caching $\widehat{k}_r^o$ dividing them by $\widehat{k}_r^{max}$, in order to increase the object caching probability. Then, we assign a value to the suboptimal binary variables $\bar{k}_r^o$, setting them to one with a probability equal to $\widehat{k}_r^o$. As a result, the algorithm chooses the node migration $\bar{m}_r$ and object caching variables $\bar{k}_r^o$.

More specifically, if we refer to Alg. 1, the solution of the continuous relaxation of the ICN model is computed in Step 1. In Step 2, the algorithm extracts $\widehat{k}_r^{max}$, that is the largest $\widehat{k}_r^o$ value for each router. Instead, the cumulative caching probability for all the objects (i.e, the value $\sum_{\forall o \in O} \widehat{k}_r^o$) is computed in Step 3, where we also sort the routers in non-increasing order according to such metric. The overall migration costs are denoted by C. In Steps 4 and 5,

Algorithm 2: Greedy Node Migration Algorithm

Input : $mdl \Leftarrow \langle b_{i,j}, p_{i,j}, d_c^o, r_p^o, C^M, C^S, B\rangle$

Output: obj_fun

spareBudget $\Leftarrow B$;

while $\exists r \in R | \mathit{RouterMigrationSavings}(r, mdl) > 0 \wedge$ $\mathit{spareBudget} > C^M$ **do**

  $\arg\max_{r \in R}$ RouterMigrationSavings(r, mdl) ;

  r.migrate();

  cachedObjects $\Leftarrow$ r.numOfCachedObjects();

  spareBudget $\Leftarrow$ spareBudget $-C^M - C^S \cdot$ cachedObjects;

  OffPathCachingMulticommodityFlowAlgorithm(mdl);

end

the randomized caching choice is performed: the algorithm caches object $o \in O$ at router $r \in R$ with probability $\widehat{k}_r^o / \widehat{k}_r^{max}$ if and only if there exists sufficient spare budget. In Step 6 the node migration costs are added to the value of C, while in Step 7, the storage costs are included and the caching variables are set. Finally, in Step 8, we solve the ICN model by fixing the caching and migration variables.

4.2. Greedy Heuristic

In this section we extensively describe the novel greedy heuristic we propose for the planning problem.

The rationale behind the greedy heuristic is to iteratively migrate, the "*most promising*" router, considering the benefits that can be achieved if (1) traffic requests are sent on the shortest path towards the closest *publisher* (i.e, a producer or an ICN-router storing a copy of the requested content) and (2) the router caches content objects and directly serves incoming traffic requests, thus offloading the networking infrastructure. As we will extensively discuss in the numerical results (Sec. 5), the greedy heuristic outperforms the randomized rounding with respect to both the computation time, as well as the quality of the final solution computed. Our heuristic is composed of three functions: the

"Greedy Node Migration" (illustrated in Alg. 2) is the main function and it invokes the two sub-functions *"Router Migration Savings"* (illustrated in Alg. 3), and *"Off-Path Caching Multicommodity Flow"* (illustrated in Alg. 4). In the rest of this section we extensively describe these three algorithms.

Greedy Node Migration. Alg. 2 shows the steps of this procedure. In Step 1, the spare budget variable is initialized to the B value. Then, in Step 2, whenever the spare budget is larger than the router migration cost C^M, and there exist a router $r \in R$ whose migration leads to positive savings (computed with Alg. 3), we migrate one router to ICN. In particular, in Step 3 we perform the greedy choice to select the router that maximizes the savings, while in Steps 4-6 we update the spare budget according to the node migration and cache storage costs. Network flows are routed in Step 7 using the off-path caching multicommodity flow algorithm, as in Alg. 4.

Considering off-path caching has a remarkable effect on both the way we choose the "most promising" router, as well as the way we actually route the flows. In particular, in Alg. 3 we compute the cost savings that the operator can experience by migrating a given router to ICN, whereas flows are routed according to Alg. 4, as we describe below.

Router Migration Savings. The algorithm that computes the router migration savings is shown in Alg. 3. Among the input parameters we provide r, which is the router for which we want to compute the migration savings. The first steps of the algorithm are to initialize the costs and savings variables (Steps 1-2). Then, we compute the overall savings we can achieve by migrating router r to ICN. In particular, we want to store in the cache of router $r \in R$, all the objects that reduce the retrieval cost for the network operator. In order to do that, we compute in Step 4 the shortest path from consumer c to the closest publisher (either a producer, or a migrated ICN router caching object o), and in Step 5 we compute its cost. Similarly, in Steps 6-7 we compute the path and cost from c to router r, and in Step 8 we increment the cache savings if consumer c is closer to r than the original publisher it was using. After summing the benefits for all the consumers, in Steps 9-11 we choose whether it is worth to

Algorithm 3: Router Migration Savings Algorithm

Input : $r, mdl \Leftarrow \langle b_{i,j}, p_{i,j}, d_c^o, r_p^o, C^M, C^S, B \rangle$

Output: savings

1 costs $\Leftarrow C^M$;

2 savings $\Leftarrow 0$;

foreach $o \in O$ **do**

3   cacheSavings $\Leftarrow 0$;

    foreach $c \in C$ **do**

4         $sp \Leftarrow$ GetShortestPathToClosestPublisher$(c, o, p_{i,j})$;

5         $ct \Leftarrow$ GetPathCost(sp);

6         $spr \Leftarrow$ GetShortestPath$(c, r, p_{i,j})$;

7         $cr \Leftarrow$ GetPathCost(spr);

        if $cr < ct$ **then**

8   cacheSavings $\Leftarrow$ cacheSavings $+ (ct - cr) \cdot d_c^o$;

        end

    end

9   **if** $cacheSavings > C^S \wedge costs + C^S < B$ **then**

10   savings $\Leftarrow$ savings + cacheSavings;

11   costs $\Leftarrow$ costs + C^S;

    end

end

12 savings = savings-costs;

cache object o at router r, and eventually we update the value for the savings, as well as the storage costs. Finally, in Step 12 we update the actual savings, by jointly taking into account the storage as well as the nodes migration costs.

Off-Path Caching Multicommodity Flow. In Alg. 4 we route network flows considering off-path caching. For each object o requested by a consumer c, the algorithm computes the overall traffic costs by finding in Step 2 the closest publisher (either a producer, or an ICN router caching o) and forwarding the traffic demand on the shortest path towards this destination (Step 3). Link capacity conditions are enforced in Step 3: the function computes the shortest

Algorithm 4: Off-Path Caching Multicommodity Flow Algorithm

Input : $mdl \Leftarrow \langle b_{i,j}, p_{i,j}, d_c^o, r_p^o, C^M, C^S, B \rangle$

Output: trafficCost

trafficCost $\Leftarrow 0$;

foreach $c \in C$ **do**

  foreach $o \in O$ **do**

    $sp \Leftarrow$ GetShortestPathToClosestPublisher$(c, o, p_{i,j})$;

    ForwardFlowOnShortestPath(c, o, d_c^o, sp);

    trafficCost $\Leftarrow$ trafficCost + $d_c^o \cdot$ GetPathCost(sp);

  end

end

path on the residual capacity graph and allocates flows according to the spare link resources.

For the sake of completeness, we want to remark the fact that the procedure *GetPathCost* returns the cost of the path that it receives as input parameter, by summing all the $p_{i,j}$ values on the given path. Furthermore, the procedure *GetShortestPathToClosestPublisher*, receives as input parameters the consumer c, the object it is requesting o, as well as the set of link prices $p_{i,j}$; it then returns as output parameter the shortest path from c to the closest "*publisher*" that can be either a producer publishing o, or an ICN router caching a copy of the requested object.

5. Numerical Results

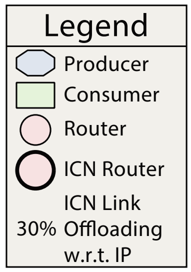
In this section we present the numerical results obtained by performing extensive analysis using our *content-aware network planning models* and the corresponding *randomized rounding* and *greedy* heuristics. In particular, in Sec. 5.1 we describe the parameters we used for the numerical analysis, while Sec. 5.2 shows the results obtained in an example scenario. In Sec. 5.3 we discuss the computational performance of the proposed algorithms. Sec. 5.4 shows the effect of the budget parameter, while Sec. 5.5 presents the sensitivity analysis to different pricing policies. Finally in Sec. 5.6 we discuss the effect of unsplittable routing conditions.

5.1. Parameters and Assumptions

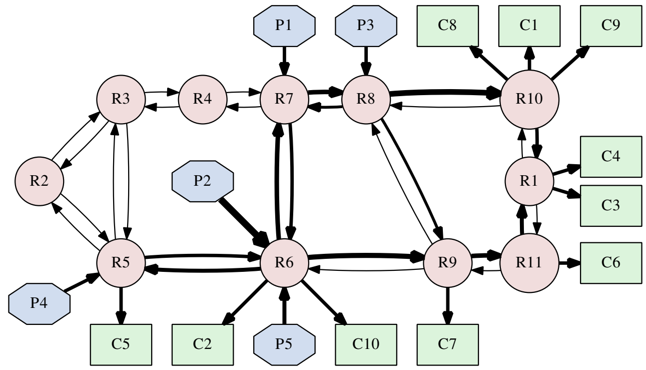
Five real topologies have been considered: Netrail (7 nodes), Abilene (11 nodes), Claranet (15 nodes), Airtel (16 nodes) and Géant (27 nodes) [26]. We uniformly distribute 5 producers and 10 consumers in the network, connecting them at most to one router. All network links have a capacity of 10 Gbit/s, and each consumer generates an aggregate demand of 1 Gbit/s randomly distributed on the object catalog according to the Zipf popularity distribution [27]. Two Zipf alpha exponents have been considered (as in [27]): $\alpha = 1.2$ is used to model a very skewed popularity distribution where few objects are frequently requested, whereas $\alpha = 0.8$ better represents less skewed demands. The object catalog is composed of 10^8 different packet *chunks* of 4kB each, as in [28, 29]. For scalability reasons, and as frequently done in the ICN literature [28], we aggregate the traffic demands on 100 popularity classes; in other words, we solve the planning problem setting $|O| = 100$. We further assume that traffic demands are expressed by the users for a mid-term timespan of one year, and thus 37 Pbytes will be transferred by the network to the consumers.

To transfer 1 Gbyte of data, Amazon nowadays charges a variable price in the range $[0.02; 0.085]$ USD [21]. Given such pricing, if 1 Gbit/s is constantly transferred on a link, its yearly cost will be in the range $[79k; 197k]$ USD; therefore, we uniformly generate the link price values ($p_{i,j}$) accordingly. Let $\max_p = 197k$ USD be the maximum yearly cost that the operator has to pay to satisfy the consumer's demand. We assume the cost to install one unit of storage is equivalent to 1/100 of the yearly traffic cost, (i.e, $C^S = 0.01 \max_p$), and similarly we set $C^M = \max_p$ for the router migration. Finally, we assume that the total migration budget B is in the range $B \in [1; 7] \max_p$, and therefore we let at most 7 nodes migrate[2]. For each analysis, we performed 50 different runs and

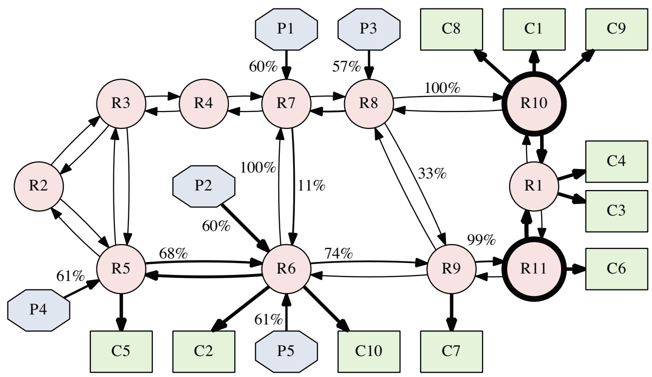
[2]This is done to make sure that at most all the nodes in the Netrail topology can migrate.



(a) Legend



(b) IP Network



(c) Optimal ICN Network

Figure 3: *Example Scenario.* We consider the Abilene topology with $\alpha = 0.8$ and budget of $B = 3.5 \max_p$. Fig. 3b represents the IP network, while Fig. 3c is the solution of the ICN planning model. The cost for the IP network is $14.3{\cdot}10^6$ USD, while in the optimal ICN network it decreases to $7.6{\cdot}10^6$ USD. Line size is proportional to the traffic the link is transferring, while the offloading percentage reported on each link in Fig. 3c compares the traffic in ICN w.r.t. the one of the IP network.

we computed the 95% confidence intervals depicted in the figures. For the sake of brevity, in this paper we present the most remarkable results, while the full set of plots is available online [30].

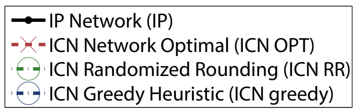
5.2. Example Scenario

Fig. 3b represents the solution obtained considering the Abilene network topology for the IP network model. Despite the fact that this result refers to a Zipf with $\alpha = 0.8$, producers have a remarkably different load; in particular the first and the second most popular objects are published by producer P2 and P5, respectively, thus their links are the most congested. Fig. 3c, instead, represents the solution for the corresponding example migrated to ICN.

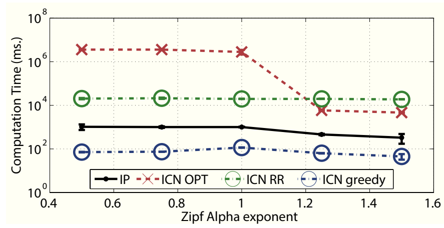
By comparing the solution depicted in Fig. 3b and Fig. 3c, we observe that migrating to ICN leads to an overall cost of $7.6 \cdot 10^6$ USD, compared to $14.3 \cdot 10^6$ USD for the IP network. Furthermore, such migration also reduces the link and producer congestion: in fact, on average, producers in ICN are providing 40% less traffic than in the IP solution. In general, network links are much less congested thanks to the presence of the two caches installed at routers R10 and R11. In addition, there exist some arcs, such as the one between R6 and R7, that are not carrying traffic anymore. Another interesting observation is that while router R10 is the one that is serving the highest number of consumers, router R11 is preferred by the model over R1. Therefore, to fully take advantage of the benefits introduced by ICN, we need to have an adequate network planning model to find the best network planning strategy. In fact, to find out the optimal migration strategy we cannot only take into account the number of consumers a router is serving, but we must have an adequate planning model such as the one proposed in this paper.

5.3. Computing Time

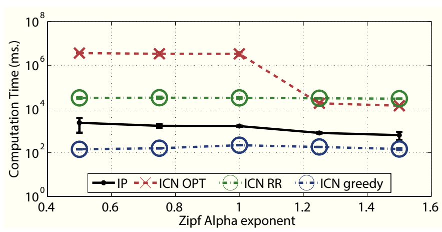
Figures 4 report the average computing time necessary to solve different instances of our planning problem using the CPLEX 12.5 solver on a Dual Intel Xeon E5-2630 v2 @ 2.60GHz machine with 64 GByte of RAM. The figures refer to the scenario with budget $B = 7 \max_p$, and the scale is logarithmic. The common legend used in all the plots we are going to discuss hereafter is reported in Fig. 4a. For the optimal ICN model (ICN OPT), we set a maximum computation time of 1 hour and a relative MIP gap tolerance between the best



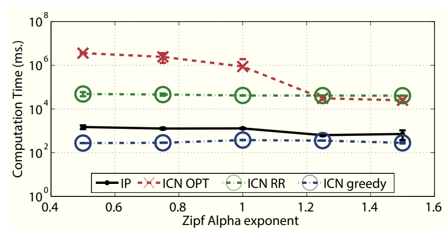
(a) Legend of the Plots



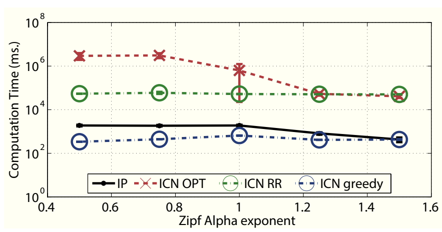
(b) Netrail ($|N| = 7$)



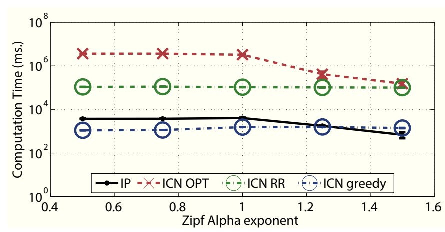
(c) Abilene ($|N| = 11$)



(d) Claranet ($|N| = 15$)



(e) Airtel ($|N| = 16$)



(f) Géant ($|N| = 27$)

Figure 4: *Computation Time Plots for Different Topologies.* Fig. 4a is the common legend we use for all the plots in Fig. 4-8. Fig. 4b-4f show the computation time for different topologies, using different algorithms, as a function of the Zipf α popularity exponent. The proposed greedy heuristic (ICN greedy) outperforms both the optimal MILP solver (ICN OPT), as well as the randomized rounding heuristic (ICN RR) cutting the execution time from hours to just few milliseconds, as shown in logarithmic scale in Fig. 4b-4f.

integer objective and the objective of the best LP relaxation to 1%. As shown in Fig. 4, the solution of ICN OPT is strongly dependent on the value of the Zipf popularity exponent α, as well as the size of the topology $|N|$. In particular, we observe that for large topologies, as well as small α values, more computation time is necessary to find the optimal solution of the planning problem. This behavior is caused by the fact that for small Zipf α exponents, the content popularity is less skewed, and therefore the object placement procedure needs to take into account also the possibility to cache objects that are not very popular. On the other hand, for large α values, the optimal solution can easily be found by the solver: the cached objects are only the very popular ones. The size of the topology $|N|$ has a similar impact on the computing time metric: the higher the number of nodes in the topology, the higher the number of combinatorial choices that the solver must take into account.

In all Figures 4b-4f there is a clear difference in computation time between the optimal solution for the ICN network obtained with the MILP solver (ICN OPT) and those obtained using the heuristics (ICN RR and ICN greedy): while the MILP solver for the optimal ICN network is very sensitive to the α parameters, the other trends are almost constant and independent from it.

We also observe that the greedy solver always outperforms of at least an order of magnitude the randomized rounding heuristic, often finding a solution in just few milliseconds rather than hours, as required by the optimal MILP solver. The greedy heuristic can scale to very large network instances, as a matter of fact, it can solve the Cogent topology (which is composed of 197 nodes), in less than 50 seconds, while instead we could not find the optimal solution using the MILP solver, in the same scenario.

Despite the fact that the computation time is a very important parameter to evaluate the performance of a heuristic algorithm, we must also consider the quality of the final solution obtained. In the following, we show that the greedy heuristic outperforms the randomized rounding even with respect to the quality of the solutions, finding a very close to optimum value, when compared to the MILP solver.

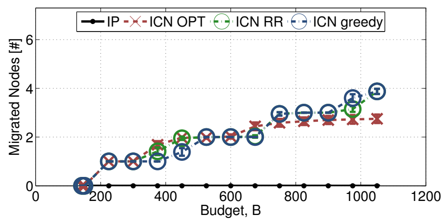
5.4. Effect of the Budget

Fig. 5-6 show the effect of the available budget for the Abilene and the Géant topology, respectively. The horizontal line represents the reference value of the IP model, where no router can migrate to ICN. The number of migrated nodes in the Abilene topology with $\alpha = 0.8$ or 1.2 is shown in Fig. 5a and 5b. For both scenarios, the randomized rounding and greedy heuristics deploy more ICN routers, especially when the available budget is large; in particular, on average, they migrate 14% more routers than the optimal solution, for $\alpha = 1.2$. In the Abilene network, at most 5 nodes are migrated to ICN, and the larger the α, the higher the number of migrated nodes. However, as shown in Fig. 5c and 5d, the amount of storage deployed is strongly dependent on the α value. In particular, on average, when $\alpha = 1.2$ the optimal solution of the ICN planning problem installs 85% less storage than for $\alpha = 0.8$. In other words, for higher α values, it is better to deploy more nodes in the network rather than increasing their storage, while the opposite holds for smaller α. Observing Fig. 5c, we notice that the greedy heuristic slightly overprovisions the caching storage installed at the nodes.

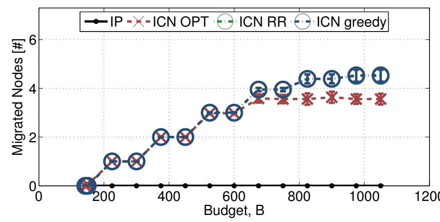
Figures 5e and 5f show the traffic cost component for the Abilene topology. In both of them there is a steep decrease in cost when the budget goes from 1 to $1.5 \max_p$. On the other hand, for a larger budget, limited improvements are observed. In terms of the quality of the computed solution, the greedy heuristic is practically overlapping the optimal solution for both Zipf α values, whereas the randomized rounding approach introduces a small approximation, and it is slightly closer to the optimum when $\alpha = 1.2$, where it finds a solution that is on average only 16% more expensive than the optimal counterpart.

Combining the key findings we observed for the computation time, as well as those reported here concerning the quality of the computed solution, we conclude that our novel greedy heuristic is to be preferred to the randomized rounding with respect to both the quality of the solution as well as the computation time.

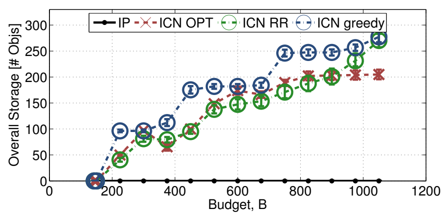
The Zipf popularity exponent has a negligible impact on the cost of the IP network, since it only affects traffic demands for single objects, but not their aggregate value. On the other, as expected, the network topology (in particular its diameter) has an remarkable effect on the overall costs, in fact, by comparing the overall cost of the IP network in the two topologies, we can conclude that, on average, Géant leads to a solution 10% more expensive than Abilene. This difference is even more remarkable, especially when considering smaller topologies;



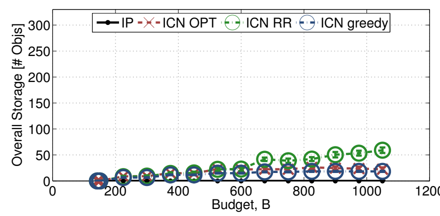
(a) Migrated Nodes $\alpha = 0.8$



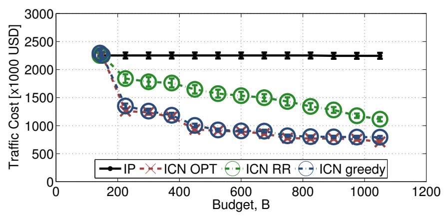
(b) Migrated Nodes $\alpha = 1.2$



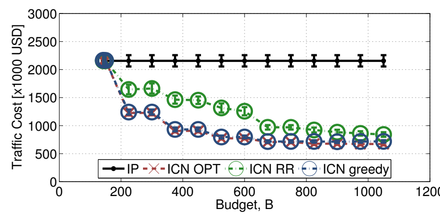
(c) Installed Storage $\alpha = 0.8$



(d) Installed Storage $\alpha = 1.2$



(e) Overall Traffic $\alpha = 0.8$

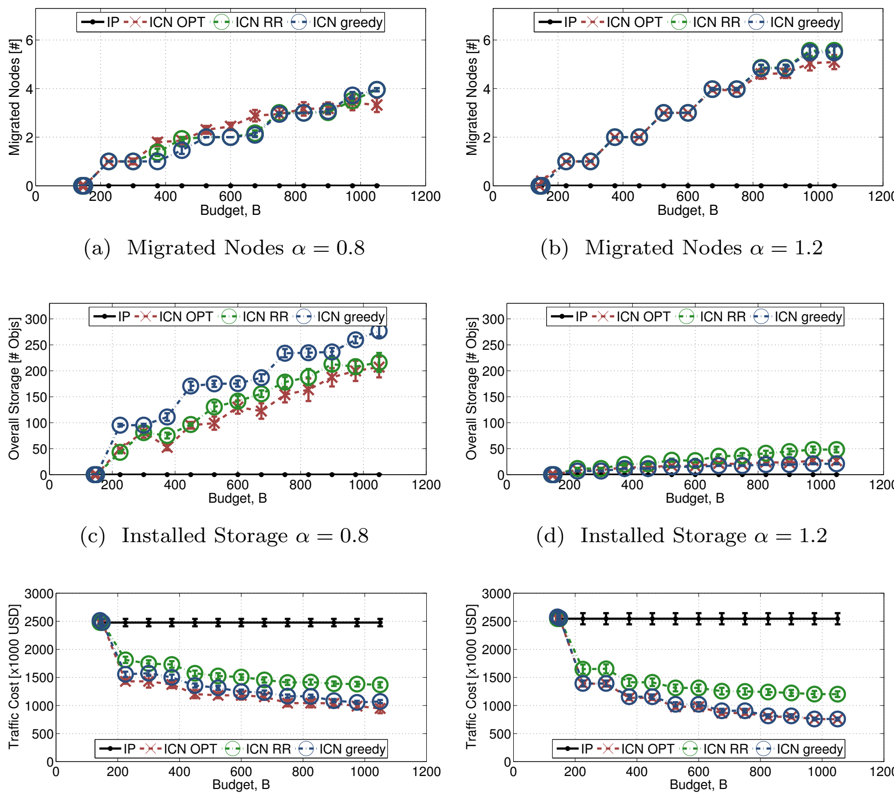


(f) Overall Traffic $\alpha = 1.2$

Figure 5: *Budget Plots, Abilene Topology.* Fig. 5a-5b show the number of migrated nodes in the Abilene topology, as a function of the migration budget, B. In Fig. 5c-5d, we represent the amount of caching storage installed, while in Fig. 5e-5f the overall traffic cost is shown.

for instance, Géant is 48% more expensive than Netrail, as shown in the full set of plots [30]. In Géant, when $\alpha = 0.8$, the randomized rounding heuristic leads to solutions which are, on average, only 16% more expensive than their optimal counterparts. However, the greedy heuristic can do even better, lowering this gap to less than 5%. In line with previous literature [22, 23], ICN allows the operator to reduce his traffic costs remarkably, even when the migration budget is very constrained, saving up to 68% of the overall traffic costs, as we observed in Géant, with $\alpha = 1.2$.

For the sake of completeness, Figures 7 show the effect of the budget in



(a) Migrated Nodes $\alpha = 0.8$

(b) Migrated Nodes $\alpha = 1.2$

(c) Installed Storage $\alpha = 0.8$

(d) Installed Storage $\alpha = 1.2$

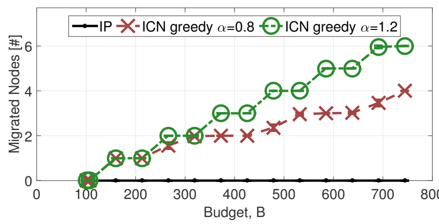
(e) Overall Traffic $\alpha = 0.8$

(f) Overall Traffic $\alpha = 1.2$

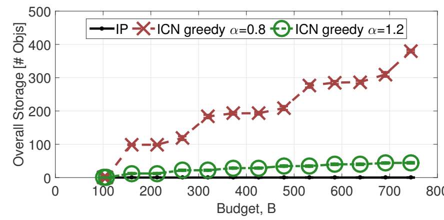
Figure 6: *Budget Plots, Géant Topology.* Fig. 6a-6b show the number of migrated nodes in the Géant topology, as a function of the migration budget. In Fig. 6c-6d, we represent the amount of caching storage installed, while in Fig. 6e-6f the overall traffic cost is shown.

the Cogent topology. Since this topology is composed of 197 nodes, we could only run the model for the IP network and our proposed heuristic for ICN. Consistently with the trends obtained in the other topologies, in Fig. 7a we observe that when $\alpha = 1.2$ a higher number of nodes is migrated to ICN, although the amount of storage deployed is an order of magnitude smaller than the one used when $\alpha = 0.8$, as shown in Fig. 7b.

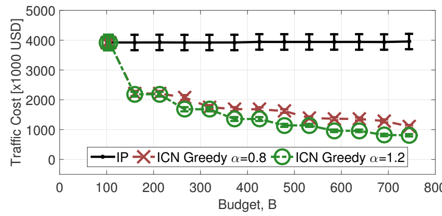
In terms of the overall traffic generated, as shown in Fig. 7c, in the Cogent topology we observe a similar trend to the one of Abilene and Géant. As expected, since the number of network nodes in Cogent is higher than that of



(a) Migrated Nodes



(b) Installed Storage



(c) Overall Traffic

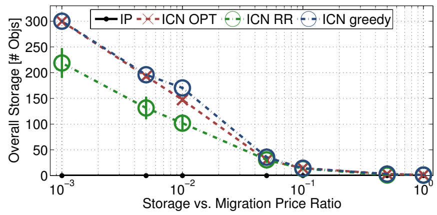
Figure 7: *Budget Plots, Cogent Topology.* Fig. 7a shows the number of migrated nodes in the Cogent topology, as a function of the migration budget. In Fig. 7b, we represent the amount of caching storage installed, while in Fig. 7c the overall traffic cost is shown.

Géant and Abilene, the overall traffic cost for the IP network is 38% higher than Géant, and 45% higher than Abilene. Furthermore, when considering the largest budget possible, the ICN solution in Cogent is 23% and 33% more expensive (in terms of traffic costs) than Géant and Abilene, respectively.

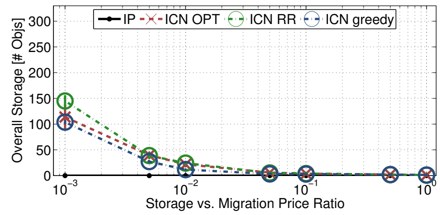
5.5. *Effect of the Price*

Figures 8 show the effect of different pricing policies for different topologies and Zipf α values. On the x-axis in Fig. 8 we report the *storage vs. migration price ratio* which is defined as $\frac{C^S}{C^M}$, where we assumed that $C^M = \max_p$ is fixed.

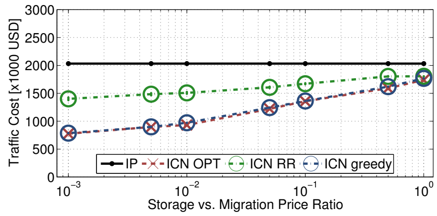
Fig. 8a-8b report the storage trend as a function of the price ratio with respect to $\alpha = 0.8$ and $\alpha = 1.2$, respectively. First of all, in both figures the trend is decreasing as the price per unit of storage increases. Furthermore, the α parameter determines the number of objects that are worth to be cached in the network. In particular, the higher the alpha, the lower their number: $\alpha = 1.2$ demands on average 60% less objects than $\alpha = 0.8$.



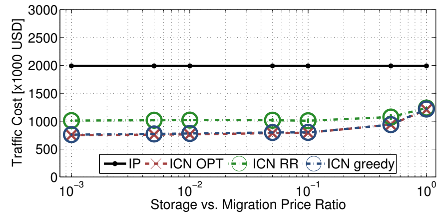
(a) Abilene Installed Storage $\alpha = 0.8$



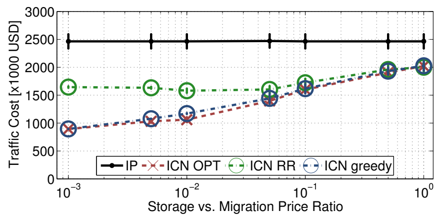
(b) Abilene Installed Storage $\alpha = 1.2$



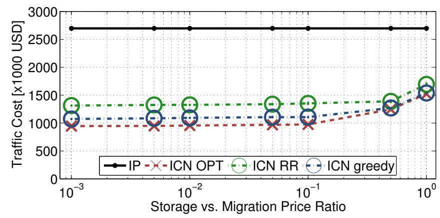
(c) Abilene Overall Traffic $\alpha = 0.8$



(d) Abilene Overall Traffic $\alpha = 1.2$



(e) Géant Overall Traffic $\alpha = 0.8$



(f) Géant Overall Traffic $\alpha = 1.2$

Figure 8: *Price Plots.* In Fig. 8a-8f we show the sensitivity to different pricing policies for the migration to ICN. Fig. 8a-8b refer to the amount of storage installed in the Abilene topology, while in Fig. 8c-8f we represent the overall traffic costs for the Géant and Abilene topologies, as a function of the *storage vs. migration price ratio*, which is defined as $\frac{C^S}{C^M}$, where the C^M value is fixed to $C^M = \max_p$.

The price sensitivity with respect to the total traffic is instead depicted in Fig. 8c-8d for the Abilene topology, and in Fig. 8e-8f for Géant. While considering $\alpha = 0.8$ as in Fig. 8c and 8e we observe that even when the price ratio is very competitive (and equal to 10^{-2}), the randomized rounding heuristic only finds a suboptimal solution, whereas the greedy heuristic is very close to the optimum.

Another interesting observation is that when we consider a higher Zipf $\alpha =$

1.2 as in Fig. 8d and 8f, the effect of the different price ratio become instead negligible, for a very large span of values. This observation is remarkable, especially when considering the real benefits that a future deployment of this technology may achieve: for skewed popularity distributions, the pricing policy has only marginal effects.

5.6. Effect of Unsplittable Routing

Our proposed optimization model for the IP network enforces unsplittable routing conditions, whereas in the ICN formulation, only ICN-migrated routers can split flows on multiple paths. In this subsection we relax both these assumptions and consider the *splittable* scenario, according to which network nodes can route a single flow on multiple paths.

In Tables 2-3 we show the results we obtained comparing the performance of splittable and unsplittable routing for IP networks. In particular, in Table 2 we show the average values, while in Table 3 we show the maximum results we obtained in terms of the traffic increase that a splittable IP network can accommodate, as well as the cost savings experienced.

The rationale behind the choice of these metrics is that, in the chosen instances of our network model, by letting intermediate nodes support splittable routing we increase the size of the feasibility region making the network forward up to 50% more traffic than the traffic it can deliver in the unsplittable scenario (Airtel topology, Table 3). On the other hand, in terms of cost benefits, splittable routing leads to very modest cost savings.

By comparing the results provided in Table 2 with those in Table 3 we can conclude that there exist a large gap between the values observed on average, and those experienced in the worst case scenarios. In particular, there are some instances in which splittable routing can significantly improve the performance, reducing costs up to 12% (Netrail topology, Table 3). However, the average cost benefits accountable to splittable routing are negligible, and lead to savings up to 2%, considering the average values (Table 2). Finally, we observe that most of the benefits can be obtained when the content popularity is more skewed, as

Table 2: Splittable and unsplittable routing, IP network, average values.

Topology	**Average Traffic Increase**		**Average Cost Decrease**	
	$\alpha = 0.8$	$\alpha = 1.2$	$\alpha = 0.8$	$\alpha = 1.2$
Abilene	3%	7%	2%	$< 1\%$
Airtel	1.2%	3.4%	$< 1\%$	$< 1\%$
Claranet	1.5%	5%	$< 1\%$	$< 1\%$
Géant	4%	8%	1%	$< 1\%$
Netrail	4.5%	10%	2%	1%

Table 3: Splittable and unsplittable routing, IP network, max values.

Topology	**Max Traffic Increase**		**Max Cost Decrease**	
	$\alpha = 0.8$	$\alpha = 1.2$	$\alpha = 0.8$	$\alpha = 1.2$
Abilene	13.5%	20%	10.7%	10%
Airtel	19%	50%	6%	4.5%
Claranet	27.5%	28.6%	7%	9.6%
Géant	31%	77%	8%	6.8%
Netrail	28%	43%	8%	12%

in the case where the Zipf alpha exponent is larger.

We considered the impact of splittable routing also in our models for the ICN network, and we discovered that it has a negligible impact on the final solution, leading to benefits accountable to much less than 1% with respect to both the traffic increase and cost decrease metrics. This behavior is caused by the fact that the impact of content caching dramatically pushes the network content distribution capabilities to its boundary, making it extremely challenging to further improve the network performance.

6. Related Work

In this section we survey relevant literature on Information-Centric Networks, comparing our contribution with related works. In particular, in Sec. 6.1 we illustrate the state of the art for ICN routers, while in Sec. 6.2, we report available results for off-path caching. Finally, we conclude in Sec. 6.3 discussing related works on optimal cache placement and request routing.

6.1. ICN-capable Routers

ICN-capable routers are beginning to appear, and some prototypes, peaking the remarkable throughput of 12 Tbps [11], have already been presented by Alcatel [9], Cisco [10] and Parc [11]. However, the design of these devices demands for specific hardware and software solutions to make them operate at wire speed, and these strict requirements will likely have remarkable effects on the pricing of the equipment.

A first investigation on the possible architecture of an ICN router, with special attention towards computational issues related to the content store, has been originally provided by Arianfar et al. in [31]. Perino et al. have further complemented such analysis by presenting in [32] clear quantitative insights on the memory technologies that can be used to make wire-speed processing of ICN packets a reality. In both these works, preliminary economic data (especially related to the prices of memories) have been provided. Nevertheless, it is really hard to predict the actual price of these devices, as well as the costs for their storage, since their pricing will likely mostly depend on strategic business decisions, rather than being only affected by the production costs. Given these observations, in the numerical analysis of Sec. 5 we took into account many different pricing policies, by changing the *storage vs. migration price ratio*, and studying the corresponding sensitivity for the different performance metrics we considered.

6.2. Off-Path Caching

Few works [33, 34, 35, 36] have also considered off-path caching to evaluate the benefits that can be obtained by diverting traffic requests on paths other than the direct one to the original content producer.

While improving the overall cache hit rate, these techniques often introduce non negligible overhead to exchange additional information regarding the state of local caches [33]. A preliminary analysis of the benefits of off-path caching in ICN has been provided by Draxler and Karl in [34]. By implementing off-path caching significant benefits in terms of hit-rate can be experienced compared to the basic solution of on-path caching. However, the work in [34] only considers a tree topology and does not take into account the optimal scenarios to perform a comparison with the best theoretical performance bound that can be obtained in one such network.

Barakat et al. evaluate in [35] the optimal performance that off-path caching can achieve, furthermore they also propose to reduce the communication burden that cooperative caching techniques induce, by leveraging hash functions. However their formulation only considers the object placement, whereas the cache provisioning is assumed to be an input parameter of their approach. Also Saha et al. tackled in [36] the communication overhead of cooperative caching, and their proposal can improve up to 20% server offloading, when compared to on-path caching.

Our formulation differs from previous works since we study the optimal *cache provisioning* and object placement that can be achieved in an ICN, thus we consider the best performance bounds that one such type of network can achieve, in the presence of optimal off-path caching conditions.

6.3. Optimal Cache Placement

Another branch of research is devoted to optimal cache placement and request routing for in-network content dissemination.

A pioneering work by Krishnan et al. is presented in [37], and deals with cache placement in a TCP/IP network to minimize the overall network flow. Among the key-features of their formulation, we mention that they bound the number of caches that can be installed, moreover they assume the average *flow hit-rate* is given as an input parameter. Wang et al. formulate in [22] a model to solve a storage constrained cache allocation problem with optimal object placement in ICN. They focus the analysis on discovering which parameters mostly affect the location of caches in the topology. In [38], Hasan et al. tackle the problem of minimizing the overall cost for inter-Autonomous System cache deployments in transit ISP networks, considering the server, energy and bandwidth prices. Finally, optimal content-oriented request routing is investigated by Mihara et al. in [39]. They minimize the overall traffic on the most congested link, however caching is not considered in their analytic framework.

In [40] Chai et al. present a cache decision policy for Information-Centric Networking based on the measure of the betweenness-centrality. Rather than supporting ubiquitous in-network caching, in their proposal objects tend to be cached in preferential (central) positions in the topology. In [41], Fayazbakhsh et al. claim that the benefits introduced in ICN by pervasive caching and nearest-replica routing can be obtained with an adequate CDN infrastructure. In particular this option can dramatically reduce the costs to migrate to ICN. However, rather than considering the best solution of the problems, the authors instead take into account fixed cache placement techniques as well as cache provisioning solutions and do not instead look for the best allocation possible.

Psaras et al. formulate in [42] and [43] *ProbCache* a caching scheme for information centric networks, specifically tailored to reduce cache redundancy, without requiring centralized supervision nor explicit coordination between network nodes. The authors study the performance of their proposed policy under both homogeneous as well as heterogeneous cache sizes. In particular, in [43], rather than performing optimal heterogeneous cache placement, the authors only conjecture two heuristic cache allocation strategies, showing that it is better to have heterogeneous cache deployments closer to the edge of the network.

Our MILP formulation differs from previous works for the following reasons: (1) we accurately model link capacities and traffic flows, (2) we explicitly take into account the contents (i.e, the objects), (3) we adopt an economic perspective on the subject, solving the network planning problem for the migration to an ICN and (4) we jointly solve the optimal *request routing*, *cache provisioning* and *object placement* problems in a budget-constrained scenario.

7. Conclusion

In this paper we tackled the *content-aware network planning* problem for the *migration* to an ICN, in a *budget constrained* scenario. In order to derive the optimal strategy that the operator should pursue, we formulated a Mixed Integer Linear Programming model that can be used to jointly identify the node migration strategy, with optimal object placement and request routing. Our proposed optimization model takes into account economic parameters related to: (1) the traffic, (2) the router migration and (3) the caching storage costs. We further complemented our contribution by extending our previous works with the design of a novel greedy heuristic that can solve the planning problem cutting the computation time of an order of magnitude and finding much better solutions than those computed with our previous randomized rounding heuristic.

We discovered that, by migrating only few nodes to ICN, the operator can experience up to a 68% reduction in traffic costs, compared to those of an IP network, as we observed for the Géant topology. On top of that, when the content popularity distribution is very skewed (i.e, $\alpha = 1.2$) the migrated nodes have on average 87% less storage than the one deployed when setting $\alpha = 0.8$. We also observed that the migration prices have a modest impact on the overall migration costs, and migrating to ICN leads to significant benefits for a large span of migration prices. Numerical results show that our proposed greedy heuristic can compute solutions practically overlapping the optimal one for the Abilene topology, and on average only 5% costlier than optimum in the Géant topology, while reducing the computation time to just few milliseconds even for the large topologies we took into account.

Acknowledgment

The authors would like to sincerely thank Prof. Federico Malucelli for his support in the formulation of the optimization models, as well as his contribution to our previous work [16]. This work was partially supported by French ANR in the framework of the Green-Dyspan project.

[1] Sandvine Inc. ULC, Global internet phenomena report (1st half 2014).

[2] Cisco, Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018, `http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html`, Last accessed: Jun. 2015.

[3] E. Nygren, R. K. Sitaraman, J. Sun, The Akamai network: a platform for high-performance Internet applications, ACM SIGOPS Operating Systems Review 44 (3) (July 2010) 2–19.

[4] A. Passarella, A survey on content-centric technologies for the current Internet: CDN and P2P solutions, Computer Communications 35 (1) (2012) 1–32.

[5] G. Carofiglio, G. Morabito, L. Muscariello, I. Solis, M. Varvello, From content delivery today to information centric networking, Computer Networks 57 (16) (2013) 3116 – 3127.

[6] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, A survey of information-centric networking, IEEE Communications Magazine 50 (7) (2012) 26–36.

[7] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, R. L. Braynard, Networking Named Content, in: Proc. of the 5th Int.l Conf. on Emerging networking experiments and technologies (CoNEXT), ACM, Rome, Italy, December 2009, pp. 1–12.

[8] Y. Kim, I. Yeom, Performance Analysis of In-Network Caching for Content-Centric Networking, Computer Networks 57 (13) (2013) 2465–2482.

[9] D. Perino, M. Varvello, L. Linguaglossa, R. Laufer, R. Boislaigue, Caesar: A Content Router for High-speed Forwarding on Content Names, in: Proc. of the 10th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, ANCS '14, 2014, pp. 137–148.

[10] W. So, A. Narayanan, D. Oran, M. Stapp, Named Data Networking on a Router: Forwarding at 20Gbps and Beyond, in: Proc. of the ACM SIGCOMM, Hong Kong, China, Aug. 2013, pp. 495–496.

[11] CCNX Community Meeting (CCNxCon 2013) Keynote, Palo Alto, CA, USA, Sep. 2013. Speaker: Glenn Edens - http://www.ccnx.org/events/ccnxcon-2013/ Last accessed: Jun. 2015.

[12] G. Pallis, Improving Content Delivery by Exploiting the Utility of CDN Servers, in: Proc. of the 5th Int.l Conf. on Data Management in Cloud, Grid and P2P Systems (Globe), LNCS, Springer, Vienna, Austria, September 5-6, 2012, pp. 88–99.

[13] L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker, Web caching and Zipf-like distributions: Evidence and implications, in: Proc. of IEEE INFOCOM, Vol. 1, New York, USA, March '99, pp. 126–134.

[14] G. Zhang, Y. Li, T. Lin, Caching in information centric networking: A survey, Computer Networks 57 (16) (2013) 3128–3141.

[15] M. Garmehi, M. Analoui, M. Pathan, R. Buyya, An economic replica placement mechanism for streaming content distribution in Hybrid CDN-P2P networks, Computer Communications 52 (2014) 60–70.

[16] M. Mangili, F. Martignon, A. Capone, F. Malucelli, Content-Aware Planning Models for Information-Centric Networking, Proc. of IEEE Global Communications Conference, Globecom 2014).

[17] B. Fortz, M. Thorup, Internet traffic engineering by optimizing OSPF weights, in: IEEE INFOCOM, Vol. 2, IEEE, Tel Aviv, Israel, 2000, pp. 519–528. doi:10.1109/INFCOM.2000.832225.

[18] W. Ben-Ameur, E. Gourdin, Internet Routing and Related Topology Issues, SIAM Journal on Discrete Mathematics 17 (1) (2003) 18–49. doi:10.1137/S0895480100377428.

[19] A. Bley, Approximability of unsplittable shortest path routing problems, Networks 54 (1) (2009) 23–46. doi:10.1002/net.20303.

[20] A. Altin, B. Fortz, M. Thorup, H. Ümit, Intra-domain traffic engineering with shortest path routing protocols, Annals of Operations Research 7 (4) (2013) 301–335. doi:10.1007/s10288-009-0113-0.

[21] Amazon CloudFront Pricing, Available at: http://aws.amazon.com/cloudfront/pricing/ Last accessed: Jun. 2015.

[22] Y. Wang, Z. Li, G. Tyson, S. Uhlig, G. Xie, Optimal Cache Allocation for Content-Centric Networking, in: Proc. of 21st IEEE ICNP, Göttingen, Germany, 7-10 October, 2013, pp. 1–10.

[23] M. Mangili, F. Martignon, A. Capone, A Comparative Study of Content-Centric and Content-Distribution Networks: Performance and Bounds, Proc. of the IEEE Global Communications Conference (GLOBECOM) (Atlanta, GA, USA, December 2013) 1425–1431.

[24] S. G. Kolliopoulos, C. Stein, Approximation Algorithms for Single-Source Unsplittable Flow, SIAM Journal on Computing 31 (3) (2001) 919–946.

[25] O. Klopfenstein, A Randomized Rounding Heuristic to Reroute Tunnels in MPLS Networks, in: Proc. of the 5th Int.l Workshop on Design of Reliable Communication Networks, (DRCN), Ischia, Naples, Italy, Oct. 2005, pp. 461–468.

[26] The Internet Topology Zoo, http://www.topology-zoo.org/ Last accessed: Jun. 2015.

[27] C. Fricker, P. Robert, J. Roberts, A Versatile and Accurate Approximation for LRU Cache Performance, in: Proc. of the 24th International Teletraffic Congress, International Teletraffic Congress, Krakow, Poland September 4-7, 2012, p. 8.

[28] G. Carofiglio, V. Gehlen, D. Perino, Experimental Evaluation of Memory Management in Content-Centric Networking, in: Proc. of IEEE Int.l Conf. on Communications (ICC), Kyoto, Japan, June 2011, pp. 1–6.

[29] C. Fricker, P. Robert, J. Roberts, N. Sbihi, Impact of traffic mix on caching performance in a content-centric network, in: Proc. of IEEE INFOCOM Workshops, Orlando, FL, USA, March 2012, pp. 310–315.

[30] Complete Set of Plots and References to the Considered Topologies, Available at: `https://www.lri.fr/~mangili/heuristics-icn.html` Last accessed: Jun. 2015.

[31] S. Arianfar, P. Nikander, J. Ott, On content-centric router design and implications, in: Proc. of the Re-Architecting the Internet Workshop (ReArch), ACM, Philadelphia, PA, USA, Nov. 2010, p. 5.

[32] D. Perino, M. Varvello, A Reality Check for Content Centric Networking, in: Proc. of the ACM SIGCOMM Workshop on Information-centric Networking (ICN), Toronto, Ontario, Canada, Aug. 2011.

[33] G. Xylomenos, X. Vasilakos, C. Tsilopoulos, V. A. Siris, G. C. Polyzos, Caching and Mobility Support in a Publish-Subscribe Internet Architecture, IEEE Communications Magazine 50 (7) (2012) 52–58.

[34] M. Draxler, H. Karl, Efficiency of On-Path and Off-Path Caching Strategies in Information Centric Networks, in: Proc. of IEEE Int.l Conf. on Green Computing and Communications (GreenCom), IEEE, 2012, pp. 581–587.

[35] C. Barakat, A. Kalla, D. Saucez, T. Turletti, Minimizing Bandwidth on Peering Links with Deflection in Named Data Networking, in: Proc. of the 3rd Intl. Conf. on Communications and Information Technology (ICCIT), IEEE, Beirut, Lebanon, June 2013, pp. 88–92.

[36] S. Saha, A. Lukyanenko, A. Yla-Jaaski, Cooperative Caching Through Routing Control in Information-Centric Networks, in: Proc. of IEEE INFOCOM, Turin, Italy, April 2013, pp. 100–104.

[37] P. Krishnan, D. Raz, Y. Shavitt, The Cache Location Problem, IEEE/ACM Transactions on Networking 8 (5) 568–582.

[38] S. Hasan, S. Gorinsky, C. Dovrolis, R. K. Sitaraman, Trade-offs in Optimizing the Cache Deployments of CDNs, in: Proc. of the 33rd IEEE Int.l Conf. on Computer Communications (INFOCOM), Toronto, Canada, May 2014.

[39] H. Mihara, D. Imachi, M. Yamamoto, T. Miyamura, K. Sasayama, Content Aware Routing: A Content Oriented Traffic Engineering, in: Proc. of the IEEE Global Communications Conference (GLOBECOM), Atlanta, GA, USA, December 2013, pp. 1438–1443.

[40] W. K. Chai, D. He, I. Psaras, G. Pavlou, Cache "less for more" in information-centric networks (extended version), Computer Communications 36 (7) (2013) 758–770.

[41] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, S. Shenker, Less Pain, Most of the Gain: Incrementally Deployable ICN, in: Proc. of ACM SIGCOMM, Hong Kong, China, August 2013, pp. 147–158.

[42] I. Psaras, W. K. Chai, G. Pavlou, In-Network Cache Management and Resource Allocation for Information-Centric Networks, IEEE Transactions on Parallel and Distributed Systems 99 (Preprint) (2014) 1.

[43] I. Psaras, W. K. Chai, G. Pavlou, Probabilistic In-network Caching for Information-centric Networks, in: Proc. of the 2nd ACM SIGCOMM Workshop on Information-centric Networking (ICN), Helsinki, Finland, Aug. 2012, pp. 55–60.