On Content Indexing for Off-Path Caching in Information-Centric Networks*

Suzan Bayhan
University of Helsinki, Finland
bayhan@hiit.fi

Liang Wang
University of Cambridge, UK
lw525@cam.ac.uk

Jörg Ott
Technische Universität München, Germany
ott@in.tum.de

Jussi Kangasharju
University of Helsinki, Finland
jakangas@helsinki.fi

Arjuna Sathiaseelan
University of Cambridge, UK
as2330@cam.ac.uk

Jon Crowcroft
University of Cambridge, UK
jac22@cam.ac.uk

ABSTRACT

A name resolution server (NRS) in an Information-Centric Network can leverage off-path copies in the network, which may not be accessible via content discovery mechanisms. Such capability is essential for an Autonomous System (AS) to avoid the costly inter-AS traffic for external content, to yield higher bandwidth efficiency for intra-AS traffic, and to decrease the data access latency for a pleasant user experience. However, these benefits come at the expense of storage and NRS update costs, for which scalability is paramount given huge number of contents. In this article, we argue that most of the benefits of an NRS can be realized by indexing only a small fraction of the requested contents benefiting from the NRS the most. First, we model the cost of serving each content in the existence of an NRS and lack of it, considering content's popularity, availability, size, and type. Next, we derive the optimal indexing decision under a given NRS size constraint by an optimization problem that minimizes total cost for serving all requests within this AS. Our results suggest that an NRS tracking even only a tiny fraction of the most popular (external) content delivers most of the benefits of an NRS, e.g., lower inter-AS traffic, higher cache hit, and lower latency. While larger NRS provides slightly higher cache hits for small caches, the impact is more visible for larger cache capacity. In contrast to diminishing gains in cache hit, data latency decreases further with increasing NRS size owing to faster name resolution.

CCS Concepts

•**Networks** → **Network architectures;** *Network performance modeling;* Network layer protocols;

Keywords

Information-Centric Networks; Name resolution; Content delivery; Scoped-flooding

1. INTRODUCTION

A key strength of an Information-Centric Network (ICN) lies in its capability to leverage the content copies stored temporarily in the in-network caches.[1] However, locating these copies is challenging due to the difficulty of deploying a scalable name resolution scheme. Current ICN architectures apply either *name resolution service* which implements a logically centralized *lookup-by-name* approach, e.g. the rendezvous point in the PSIRP/PURSUIT architecture [11], or *route-by-name* approaches, e.g. NDN/CCNx [27], or a combination of both, e.g. as in SCANDEX [20].

In approaches that involve a standalone service for name resolution, content publishers register their content with the responsible network entity, e.g., resolution handlers in DONA [14], rendezvous point in PSIRP, or name resolution service in NetInf [2]. Registration messages propagate in the network depending on the underlying name resolution architecture and defined policies, e.g., messages may be forwarded to peering Autonomous Systems (ASes) or/and to parent AS in the AS hierarchy. In lookup-by-name, requesting router first consults a directory service, e.g. acting as DNS in Internet, to retrieve the (nearest) locator for the requested content. We refer to this directory service as *Name Resolution Server* (NRS). In other words, *name resolution* is the first step in content delivery. After the name is resolved to a locator by the NRS, the requesting router initiates *request routing* towards the content provider, and triggers the *content retrieval* with the discovered content provider. In route-by-name, there is no dedicated directory service. Instead, name resolution is coupled with request routing. Routers propagate the content request messages using the content name through the related entities until reaching the node with the knowledge of at least one content provider [4].

While the scalability of the NRS has raised concerns [8,9,13,17], recent work [21] has shown that ICN entities, e.g., forwarding information base, can scale up to billions of entries using efficient data structures and applying speculative forwarding. An NRS designed with a similar approach can leverage a temporary copy of the content that resides in an in-network cache and closer to the requesting router than the origin server hosting a permanent copy. In the lack of an NRS, the inefficiency in discovering the temporary copies leads to three types of inefficiencies:

(i) **Bandwidth inefficiency**: If the content is a local content, i.e., publisher of the content is also in the same AS, missing a temporary copy in the vicinity results in bandwidth inefficiency (which can also be represented as monetary inefficiency.).

(ii) **Monetary inefficiency**: If the content is not local, it must be retrieved from external ASes, which has a more direct representation of monetary cost as inter-AS traffic is subject to charges based on traffic volume.[2] In the lack of an NRS, on-path hits or off-path content search helps discovering the nearby copies, however the former's contribution may be limited [10] whereas the latter has to

*to appear in ACM ICN 2016.

[1]We use *in-network cache*, *node*, and *router* interchangeably in the rest of the article.

[2]Business relations between ASes define this monetary cost. ASes may peer with others and carry each other's traffic without any cost.

balance the cost of discovery vs. the chance of hitting a content copy [6, 23].

(iii) **Longer latency**: While the previous two inefficiencies are network-centric, resulting inefficiency from the user's viewpoint is longer latency and possibly poor user experience, which may eventually lead to monetary inefficiency.

An NRS which keeps track of the content locations promises to alleviate the above-listed inefficiencies. But, routers have to update the NRS upon a change in their cache states to provide an accurate view of the network state. Therefore, it is a legitimate question to ask how to tune the trade-off between the scalability of an NRS and the benefits it provides. The motivation of this paper is to explore this trade-off between the gains offered by the NRS by discovering an off-path content object that would otherwise be inaccessible and the cost of maintaining the NRS. Name resolution architecture may be hierarchical such as DONA [14] or distributed structures such as DHT-based solutions [8]. Rather than relying on a particular design, we assume a logically centralized NRS, which may implement one of these options, and focus on the benefits of accessing the cached copies via the NRS.

More particularly, we raise the following questions: *How does an NRS which stores locators for the cached copies of the content items affect bandwidth efficiency and content discovery/retrieval cost? Can we have an NRS that indexes only some of the content but brings most of the benefits?* With the goal of addressing these questions, we establish the following key contributions:

- From the viewpoint of an AS, we model for each content the discovery, retrieval, and NRS update costs, considering the type of content, i.e., whether its publisher is also in the same AS domain or not, its popularity, availability, and size (Section 2 and 3).
- We find the optimal NRS setting by formulating an optimization problem which minimizes the expected cost of serving user requests from within this AS (Section 4). As ensuring NRS scalability is crucial, we add the number of contents to be indexed as a constraint to our optimization problem. Moreover, to decrease the number of NRS update messages, we apply rate-based updates which can ensure a certain degree of NRS accuracy as proposed in [3].
- We provide a thorough analysis on the cases where the NRS benefits are maximum by solving the formulated problem using real AS topologies (Section 5). Our results suggest that an AS can increase its cache hit ratio, decrease the inter-AS traffic, and data access latency by indexing only the most popular and particularly external content, given that inter-AS traffic is costly. Moreover, improvements are more visible for larger cache regime, where the temporary content copies are more likely to exist in the vicinity; and in the lack of NRS, flooding-based content discovery and retrieval may waste bandwidth by redundantly routing multiple copies of the content.

2. SYSTEM MODEL

We consider an AS as in Fig. 1 which consists of N nodes with cache capacity L_c and an NRS storing the mapping between the content and its providers. A node n_i stores in its cache temporarily a subset of the content items from the content catalogue $C = \{c_1, \cdots, c_K\}$ of K items. Moreover, a node may act as origin server by hosting the original copy of some content in its permanent storage. We assume that there is a single origin server for each content.

Each content is characterized by its popularity (or request rate, q_k), size (s_k), and availability (α_k). A content object may be *local* content, if the content's publisher is also in the same AS domain as the content requester, and *external* otherwise. Note that local content exists permanently at the origin server(s) and may temporarily be hosted in some caches in the AS. In contrast, external content can only be stored temporarily in this AS domain.

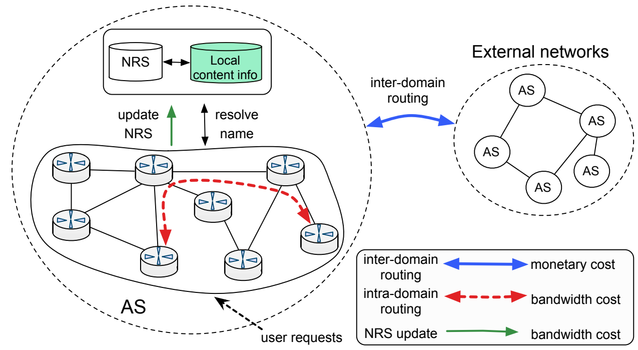


Figure 1: System model: The AS hosts an NRS for indexing the temporary content copies and a server for local publisher information, i.e., permanent copies. External networks are abstracted as a single external AS whose content can be accessed via inter-domain routing and is subject to monetary cost based on AS peering relations.

Since an AS must be aware of the content published inside its domain, we assume that it hosts a database to store local content information, e.g., content id and publisher id. We assume that the content distribution is uniform, i.e., a content item may be stored by each node equally probably. Content availability of c_k denoted by α_k is the probability that this item is expected to be available in a node's cache. Using Che's approximation [5], we calculate α_k as a function of cache capacity, content popularity, and number of items for Least-Recently Used (LRU) policy and assuming Independent Reference Model.

In our model, the NRS is a logically centralized entity and is reachable by each node in the network. Since ensuring NRS scalability is paramount, the AS takes the following two measures. First, to avoid excessive signalling overhead in the network, nodes occasionally update the NRS based on the minimum required update rate as proposed in [3] to ensure that NRS has a coherent and accurate view of the content locations. Second, to avoid the cost of indexing (e.g., signalling and look-up overheads [13]), the NRS indexes only $\omega \in [0, 1]$ fraction of the requested content items rather than storing locators for every content item. We represent an indexing scheme as a vector of binary variables $\mathbf{I} = [x_k]$ where $x_k = 1$ represents that NRS stores which nodes are the providers of c_k. Under $\mathbf{I}$ with ω, size of the NRS equals to $K_\omega = \lfloor K\omega \rfloor$. Based on $\mathbf{I}$, the NRS state for c_k can take one of the following three values:

$$S_{NRS}(k) := \begin{cases} 0, & \text{if } c_k \text{ is not in the network,} \\ 1, & \text{if } c_k \text{ is in the network,} \\ NA, & \text{if NRS does not track } c_k \text{ .} \end{cases}$$

Let $S(k)$ be a binary variable representing the existence of c_k in one of the in-network caches. We represent the system state as a tuple: $< S(k), S_{NRS}(k) >$. Depending on the periodicity of the state updates, the NRS may be outdated, i.e., $S_{NRS}(k)$ differs from $S(k)$. We define the corresponding false positives as ε^1 and false negatives as ε^0 for all $x_k = 1$. As a result, state space for our system consists of six states whose state probability is denoted by $p_{x,y}$ and defined as:

$$p_{xy} = Pr\{S(k) = x,\ S_{NRS}(k) = y\}, \quad x \in \{0, 1\}, y \in \{0, 1, NA\}.$$

Table 1 provides the probability of each state and corresponding actions taken by the entities upon a user request.

3. NRS-BASED CONTENT DELIVERY

In this section, we present how a router receiving a user request serves the requested content under a particular NRS setting and calculate the associated cost of content delivery.

3.1 Request Routing

When a request for c_k arrives to an ICN router n_i, n_i first checks its cache. There are three possibilities:

The requested content is already in n_i's cache: n_i, which we refer to as the *requesting router*, serves this request from its cache and with cost ϕ^c. In the following, we assume that the cost of one bit transmission on a link between two routers in the AS as 1, i.e., $\phi^{\text{as}} = 1$ per bit, and refer to all other costs in terms of their ratio to ϕ^{as}.

The NRS has an entry for c_k: If c_k is not in n_i's cache, n_i checks if the NRS has an entry for c_k. Table 1 lists the system state and corresponding actions triggered after a content request. For example, when c_k is not in any of the caches in this AS, i.e., $S(k) = 0$, or the NRS thinks so, i.e., $S_{NRS}(k) = 0$, the permanent copy of the content will be retrieved from the origin server. First, the NRS checks if this is a local content, and if so, retrieves the publisher id from the local content database and delivers this information to n_i. This request is finally satisfied from inside the AS via *intra-domain routing*. If the request is for an external content, *inter-domain routing* is triggered to retrieve this content from an external AS hosting c_k. While this is the right policy for $S(k) = 0$ and $S_{NRS}(k) = 0$, it is not for $S(k) = 1$ and $S_{NRS}(k) = 0$ as this AS hosts the content but the NRS state is inaccurate, i.e. *false negative*.

If the NRS stores id of the content provider n_j as the nearest copy holder, i.e., $S_{NRS}(k) = 1$, the request is routed to n_j on the shortest path determined by intra-domain routing. Requested content is fetched from n_j to n_i, and cache space is managed by LRU policy. As the NRS state is subject to *false positive*, i.e., $S(k) = 0$ and $S_{NRS}(k) = 1$, the content may not be available at n_j. Requesting router detects this case, e.g., by setting its timer to the expected round trip time which is a function of path length between the source and the target node, or by a negative acknowledgement message from n_j. If the request routing times out or a failure message is received, the content is downloaded from the content publisher which may be in the AS or outside the AS domain. We refer to this scheme as *index-based search* (IBS).

The NRS has no information on c_k: When there is no information available, n_i first checks if its neighbors have the content by sending content discovery messages. This way, a nearby off-path copy can be retrieved. While several solutions exist in the literature for content discovery, e.g., exchanging the cached content list among nodes [15, 25], we assume that nodes implement *scoped flooding* [23] due to its simplicity. The biggest hurdle of flooding-based schemes is to stop flooding *timely*, i.e., avoiding wasteful distribution of the messages but still guaranteeing some degree of success probability. The first node initiating the discovery sets the scope of flooding denoted by $h_{\max}$ by setting the maximum hop count the message could follow. We refer to this scheme as *flooding-based search* (FBS). FBS fails to discover the content in two cases: because c_k is not in the network; or c_k is in the network but the scope parameter of FBS is not set properly (please see [23] for more on tuning the scope parameter). In case of failure, c_k is fetched from the content publisher.

3.2 Content Discovery

If c_k is not in n_i's cache, n_i creates a content discovery message, which includes the id of the requested content as well as n_i's id. Resulting message size is then at least $l^{req} = \log K + \log N$. Depending on the naming scheme as well as the ICN architecture, actual message will be longer, but yet much smaller than the content. In case of an NRS entry for c_k, content discovery message is routed towards the destination. In the lack of any information, messages are sent in a pre-defined neighborhood of n_i based on FBS. Below, we present the related cost for each action.

Inter-domain routing: It is initiated only for external content and when the content i) is not, ii) is falsely believed to be not, or iii) could not be found in the AS. From the perspective of the requesting router, we can calculate the probability that c_k is in the AS, i.e., hosted by at least one of $N-1$ routers, as: $P_k = 1-(1-\alpha_k)^{N-1}$. Then, probability of (i) equals to $1-P_k$. Similarly, we calculate probability of (ii) as $P_k\varepsilon^0$. Finally, probability of (iii) equals to the probability of failure to discover the content by FBS, which we explain later. Then, expected cost of inter-domain routing is a function of the cost of inter-domain routing per bit denoted by $\phi^{\text{as-as}}$ and probabilities of (i), (ii), and (iii).

Flooding-based Search (FBS): For FBS, we need to calculate total number of nodes receiving the FBS message under hop restrictions [23]. Let N_h be the total number of nodes that are h hops away from n_i. Then, the cost of FBS with search scope set to $h_{\max}$ is expressed as:

$$\phi^{\text{FBS}} = N^{\text{FBS}} = \sum_{h=1}^{h_{\max}} N_h. \tag{1}$$

After calculating average N_h using the AS topology, we can also derive the search success γ_k which is defined as the probability that c_k is stored by at least one of the nodes within h hop distance from the requesting router. Assuming that likelihood of every node storing the requested content is the same, search success probability denoted by γ_k^{FBS} equals to:

$$\gamma_k^{\text{FBS}} = 1-(1-\alpha_k)^{N^{\text{FBS}}}.$$

Intra-domain routing: It is initiated for fetching the local content from the origin server in the AS or in case of IBS.

- *Fetching the content from the origin server in the AS*: In this case, we need to find the expected distance between a randomly picked router and a randomly picked origin server. Recall that there is only one origin server per content. Since every router is equally likely to host c_k, the expected distance to c_k equals to the average path length of the network $\bar{h}$, which depends on the network topology. Then, ϕ^{ori} denoting the expected cost of routing the request to the origin server equals to:

$$\phi^{\text{ori}} = \begin{cases} \bar{h} & \text{if local content} \\ \phi^{\text{as-as}} & \text{if external content.} \end{cases}$$

- *IBS*: We need to account for the expected hop distance between a randomly selected router and the nearest copy for the requested content, which depends on both the content availability and the network topology. Let h_k denote the expected hop distance between an arbitrary requesting router and the nearest copy of c_k. Moreover, probability that the content is discovered at exactly the h^{th} hop is p_h. Content is served from a node in the h^{th} hop only if no other node in the previous $(h-1)$ hop neighborhood has the content. Based on this fact, we calculate p_h as follows:

$$p_h = \frac{(1-\alpha_k)^{\sum_{i=1}^{h-1} N_i}(1-(1-\alpha_k)^{N_h})}{1-(1-\alpha_k)^{\sum_{i=1}^{h_{\max}} N_i}},$$

Table 1: **System state $< S(k), S_{NRS}(k) >$ and corresponding actions triggered for serving content k. IBS: Index-based search, FBS: Flooding-based search.**

		NRS state, $S_{NRS}(k)$			
		NRS indexes c_k		NRS does not index c_k	Content is retrieved from
		0	1	NA	
Content state $S(k)$	0	Fetch from the origin $p_{00}=(1-\alpha_k)(1-P_k)(1-\varepsilon^1)$	IBS, fetch from the origin $p_{01}=(1-\alpha_k)(1-P_k)\varepsilon^1$	FBS, fetch from the origin $p_{0na}=(1-\alpha_k)(1-P_k)$	AS, if local content External AS, if external content
	1	Fetch from the origin $p_{10}=(1-\alpha_k)P_k(1-\varepsilon^0)$	IBS $p_{11}=(1-\alpha_k)P_k\varepsilon^0$	FBS, fetch from the origin $p_{1na}=(1-\alpha_k)P_k$	AS, if discovered or local External AS, if external and not discovered

$$\text{Content discovery cost: } \phi_k = \begin{cases} l^{req}(\alpha_k\phi^c + (p_{11}+p_{01})\phi_k^{\text{IBS}} + (1-p_{11})\phi^{\text{ori}}), & \text{if } x_k = 1 \quad (2)\\ l^{req}(\alpha_k\phi^c + \phi^{\text{FBS}} + (p_{0na}+p_{1na}(1-\gamma_k^{\text{FBS}}))\phi^{\text{ori}}), & \text{if } x_k = 0 \quad (3)\end{cases}$$

$$\text{Content retrieval cost: } \beta_k = \begin{cases} s_k(\alpha_k\phi^c + p_{11}\phi_k^{\text{IBS}} + (1-p_{11})\phi^{\text{ori}}), & \text{if } x_k = 1 \quad (4)\\ s_k(\alpha_k\phi^c + p_{1na}\gamma_k^{\text{FBS}} n_k \phi_k^{\text{IBS}} + (p_{0na}+p_{1na}(1-\gamma_k^{\text{FBS}}))\phi^{\text{ori}}), & \text{if } x_k = 0 \quad (5)\end{cases}$$

$$\text{NRS update cost: } \psi_k = \begin{cases} R_k(\varepsilon^1,\varepsilon^0) l^{up}\phi^{\text{up}}, & \text{where } l^{up} = \log K_\omega + \log N + 1, \text{ if } x_k = 1 \quad (6)\\ 0, & \text{if } x_k = 0 \quad (7)\end{cases}$$

where $h_{\max}$ represents the network diameter. Then, the expected hop distance to the nearest content copy is:

$$h_k = \sum_{h=1}^{h_{\max}} h p_h,$$

which also equals to ϕ_k^{IBS}, the cost of IBS.

Putting all pieces together, we calculate the expected cost of content discovery for c_k as in (2) if the NRS tracks this content, and as in (3) otherwise. [3]

3.3 Content Retrieval

After a discovery message reaches a content provider, be it an in-network cache or an origin server inside or outside the AS, this provider routes the content towards the requesting router. Local content is always served from within the AS, whereas external content may sometimes be fetched from inside the AS (cf. the last column of Table 1). We calculate the cost of content retrieval β_k for c_k according to the cost of retrieving the content from the cache, from a remote cache in the AS, and from external ASes. The cost of content retrieval for c_k with size s_k, denoted by β_k, equals to (4) if the NRS tracks it, and (5) otherwise. Briefly, the first term in the parenthesis in (4) stands for the cost of getting the content from the cache, the second term for the expected cost of IBS, and the last term for the expected cost of retrieving the content directly from the origin server.

In (5), n_k is the overhead factor to account for the cases where FBS content discovery messages reach to multiple content providers, and these providers send the content in response. Any transmission other than the ones to route the nearest copy of the content is a waste of network resources. A router on the response path between the provider and the requester may stop redundant transmission related to the same content, if it has already routed a copy of the content. On the other hand, if two responses do not share any node on their path, the requester receives more than one copy of the content. Additionally, number of responses triggered by the content discovery step depends on the availability of the content in the neighborhood of the requesting router. To account for these factors, we define overhead factor n_k as:

$$n_k = 1 + \rho(N^{\text{FBS}}\alpha_k),$$

where $\rho \in [0,1]$ is *FBS redundancy coefficient*, and the term $N^{\text{FBS}}\alpha_k$ expresses the expected number of content copies in the FBS scope of the requesting router. While exact modelling of n_k and ρ is beyond the scope of this article, we note that the overhead depends on the network topology and the locations of the copies. For example, for a star topology, when the node at the centre initiates FBS and the leaves have the content, ρ will be high as none of the leaves and intermediate routers are aware of the others routing the content in parallel. Moreover, this simple model calculates the overhead as if all content copies are located at the same distance from the requesting router and thereby yielding the same IBS cost ϕ_k^{IBS} in (5). We will provide a more accurate model for the overhead in Section 5.

3.4 NRS Updates

To have an accurate view of the content locators, the NRS must be updated by the routers upon a change in their caches. However, as cache population is very dynamic, NRS updates at every cache eviction and admission may lead to a substantial overhead. Instead, routers can update the NRS with certain rate that bounds the inaccuracy at the NRS to certain limits. Azimdoost et al. [3] model the minimum rate of update required to keep the *distortion* at the NRS below certain false alarm rate ε^1 and false negative rate ε^0 using rate distortion theory. We assume that each router in the AS updates the NRS based on this scheme. For the sake of completeness, we present here the main rate equation of [3]. Let $R_k(\varepsilon^1,\varepsilon^0)$ denote the rate of update for c_k which is calculated as:

$$R_k(\varepsilon^0,\varepsilon^1) \geqslant N q_k (1-\alpha_k)\left(2 - \frac{\varepsilon^0(1-\alpha_k)}{\alpha_k(1-\alpha_k-\varepsilon^1)} - \frac{\varepsilon^1\alpha_k}{(1-\alpha_k)(\alpha_k-\varepsilon^0)}\right).$$

[3] Note that on-path caching helps decreasing content discovery cost, which we do not account for here for the sake of simplicity.

if $\varepsilon^0(1-\alpha_k)+\varepsilon^1\alpha_k < \alpha_k(1-\alpha_k)$ and $\varepsilon^0 < \alpha_k < 1-\varepsilon^1$ hold, and no update is required otherwise.

An update message must include at least the id of the router sending the update message as well as the id of the content among all K_ω items, and the new state of the content (i.e., in the cache, not in the cache). Then, the message size is at least $l^{up} = \log K_\omega + \log N + 1$. We calculate the updating cost for c_k where ϕ^{up} denotes the cost of updating per bit as in (6) if the NRS indexes this item and (7) otherwise.

4. INDEXING FOR MINIMIZING THE COST OF CONTENT DELIVERY

An AS aims to balance the trade-off between the NRS scalability and the performance gains it provides by smartly selecting only K_ω objects to be indexed from the set of all requested content objects. From the perspective of the AS, the optimal indexing scheme is the one which minimizes the expected cost of content delivery. To reflect this perspective, we define utility of c_k, denoted by U_k, in terms of the number of bits and corresponding cost of routing traffic to serve requests for c_k. The cost includes all transmissions related to discovery, retrieval, and control messages. Given our binary decision variable x_k, we calculate U_k as:

$$
\begin{aligned}
&U_k = q_k(\phi_k+\beta_k)+\psi_k\\
&\phi_k = l^{req}(\alpha_k\phi^c + (p_{11}+p_{01})\phi_k^{\text{IBS}}) + x_k((1-p_{11})\phi^{\text{ori}})\\
&\qquad +(1-x_k)(\phi^{\text{FBS}} + (p_{0na}+p_{1na}(1-\gamma_k^{\text{FBS}}))\phi^{\text{ori}}))\\
&\beta_k = \alpha_k s_k\phi^c + s_k(p_{11}\phi_k^{\text{IBS}} + (x_k((1-p_{11})\phi^{\text{ori}})\\
&+(1-x_k)(p_{1na}\gamma_k^{\text{FBS}} n_k\phi_k^{\text{IBS}}) + (p_{0na}+p_{1na}(1-\gamma_k^{\text{FBS}}))\phi^{\text{ori}}))\\
&\psi_k = x_k R_k(\varepsilon^1,\varepsilon^0) l^{up}\phi^{\text{up}}
\end{aligned}
$$

We assume that long term values of q_k and α_k are available to the AS. [4] Moreover, AS has the information on average neighborhood size, N_h for each hop value h. Then, we formulate the minimum-cost indexing scheme under the constraint that the NRS can index maximum K_ω objects as follows:

$$\min \sum_{k=1}^{K} U_k \tag{8}$$

$$s.t. \sum_{k=1}^{K} x_k \leqslant K_\omega. \tag{9}$$

Problem defined in (8)-(9) is a linear integer problem. We can equivalently represent this problem as *indexing-gain maximization problem* where we define *indexing gain* for a content item as the cost saving facilitated by the NRS for serving this content. Indexing gain for c_k denoted by ΔU_k is:

$$\Delta U_k = U_k(x_k=0) - U_k(x_k=1).$$

Then, a centralized entity located in the AS, e.g., NRS controller, can find the optimal NRS setting by selecting the first K_ω items with the highest ΔU_k value.

[4] While an AS can infer the content popularity by recording the requests for each content within its domain, in practise it is difficult to exactly know the content popularity as it changes over time. Consequently, an accurate estimation of availability is challenging. However, rather than an exact value of availability, the ordering of content objects in terms of their availabilities can in practise guide the indexing decision.

5. PERFORMANCE EVALUATION

In this section, we model our system using ICARUS [19], a publicly-available Python-based simulator, and evaluate the performance of NRS-based content delivery in an AS. For each setting, we solve the indexing-gain maximization problem and find the optimal NRS setting.

To have realistic AS topologies, we use four Rocket Fuel topologies, namely AS 3259, AS 1755, AS 3257, and AS 1239.[5] Corresponding number of routers in these ASes are $N = (79, 87, 161, 315)$; path lengths are $\bar{h} = (4.08, 4.53, 4.20, 3.97)$, and node degrees are $\bar{d} = (3.72, 3.70, 4.07, 6.17)$. For each AS, we calculate average neighborhood size at each hop, i.e., N_h, to use in our utility function. In each network, the router with the highest betweenness centrality is selected as the gateway connecting to the external AS. Unless otherwise stated, we set the cache capacity of a router as $L_c = 10^{-2}K/N$, $\varepsilon^1 = \varepsilon^0 = 10^{-2}$, and Zipf parameter to 0.8. Moreover, we assume fixed content size for all content and set it to 1 Mb to isolate the impact of size on the utility of an object. Note that this size may correspond to the first chunk of (a huge video) content which is fetched via the mechanisms introduced earlier, and the following chunks can be retrieved directly from the provider of the first chunk [6]. To represent the scenarios where inter-domain routing cost is *low*, *moderate*, and *high*, we set $\phi^{\text{as-as}} = \{3\phi^{\text{as}}, 5\phi^{\text{as}}, 10\phi^{\text{as}}\}$, respectively. Furthermore, we set $\phi^c = 0$ and $\phi^{\text{up}} = 2\phi^{\text{as}}$, and search scope to 3. These parameters can be set appropriately by the AS depending on its preferences as well as its business policy with the external ASes.

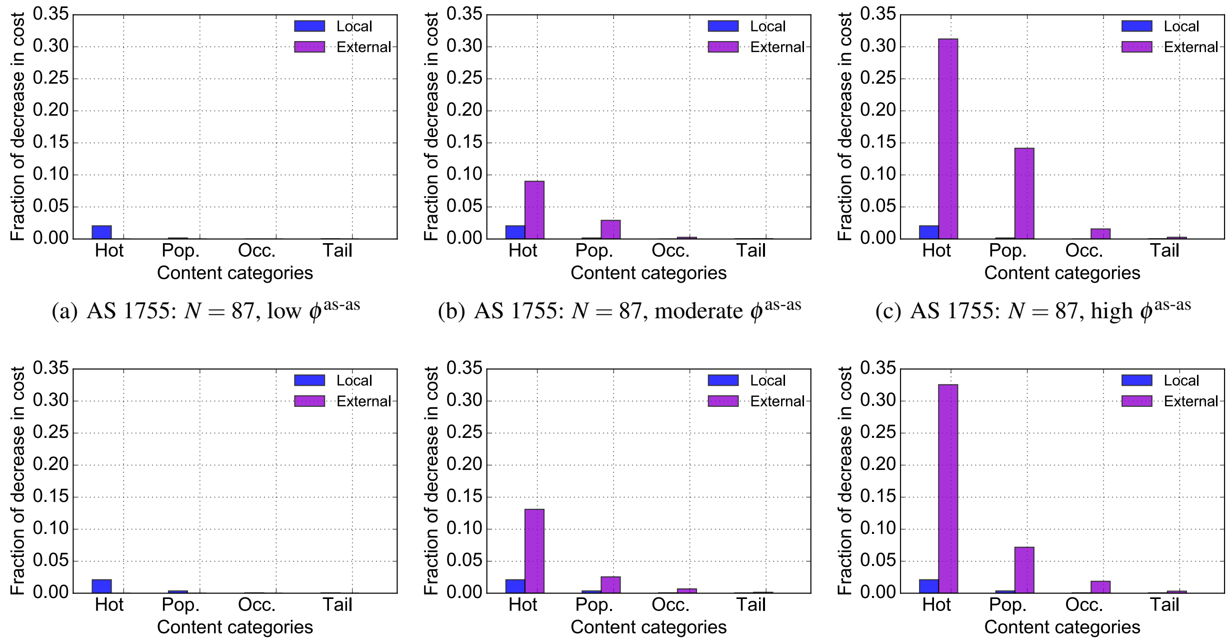
As performance measures, we define the following metrics:

- *Fraction of decrease in delivery cost* measures the effect of the NRS on a content item in terms of the fraction of decrease in that content's delivery cost. More formally, we calculate the cost reduction for c_k as $\frac{\Delta U_k}{U_k(x_k=0)}$.
- *Indexing gain* measures the expected cost savings for serving each content c_k in case the NRS indexes c_k.
- *External server hit ratio* is the fraction of requests for external content satisfied from the external AS. It reflects the inter-AS traffic.
- *Cache byte hit ratio* is the fraction of bytes served from a cache residing in the AS rather than the origin server holding the permanent copy of the content.
- *Data access latency* is the delay for serving a user request.

5.1 Impact of Inter-AS Traffic Cost ($\phi^{\text{as-as}}$)

Consider a scenario where the AS hosts 30% of the requested content from within the AS domain, i.e, fraction of local content is 0.3. Given that geography affects popularity significantly, e.g., in Turkey, content in Turkish will be more popular compared to content in Finnish, etc. [16], local content is expected to attract more attention from the users in that AS domain. On the other hand, certainly there will always be very popular items whose providers are outside the AS domain or some local content with very low popularity. To reflect these properties, we select local content from the content catalogue as follows. Using the categorization of content in [16], we divide the content catalogue into four as *hot, popular, occasional*, and *far tail*. The content objects with the highest 0.1%

[5] Some of the studied topologies in fact belong to Tier-1 AS which freely exchange traffic with other Tier-1 providers, i.e., $\phi^{\text{as-as}} = 0$. However, we use the corresponding topologies only to have a realistic AS topology.



(a) AS 1755: $N = 87$, low $\phi^{\text{as-as}}$ (b) AS 1755: $N = 87$, moderate $\phi^{\text{as-as}}$ (c) AS 1755: $N = 87$, high $\phi^{\text{as-as}}$

(d) AS 1239: $N = 315$, low $\phi^{\text{as-as}}$ (e) AS 1239: $N = 315$, moderate $\phi^{\text{as-as}}$ (f) AS 1239: $N = 315$, high $\phi^{\text{as-as}}$

Figure 2: Effect of inter-AS cost on $\Delta U_k/U_k(x_k = 0)$ for $\rho = 0$ and $s_k = 1$ Mb with various $\phi^{\text{as-as}}$ values for AS 1755 and AS 1239. Each AS hosts the 30% of the published content, i.e., 30% of the content is local content.

popularity rank are classified as hot content, 1% as popular, 10% as occasional, and the rest as far tail. The local content's popularity lies in the hot, popular, occasional, and far tail of the content popularity distribution with probabilities (0.3, 0.3, 0.2, 0.2), respectively.

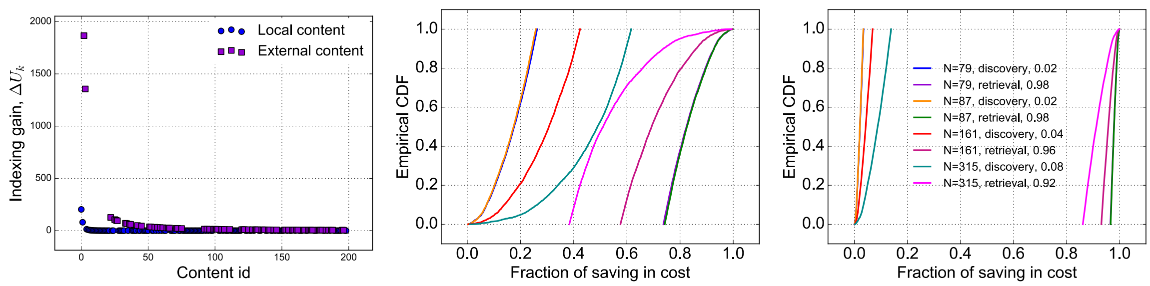
In this scenario, we set $K = 2000$. For this setting, a content discovery message is $l^{req} \approx 18-20$ bits long and an NRS update message is $l^{up} \approx 15-17$ bits long depending on the network size N. We consider two extreme values of ρ, i.e., $\rho = 0$ to represent the case where all routers are informed about the content delivery and hence only one copy of the content is routed back to the requester; and $\rho = 1$ where all copies in the search scope, i.e., $\alpha_k N^{\text{FBS}}$, are delivered to the requester. We set $\omega = 1$ to identify the related impact of the NRS on each content's delivery cost.

Fig.2 plots the fraction of decrease in cost of an item in each content class. We report results only for AS 1755 and AS 1239 as other ASes exhibit very similar behaviour. First thing we observe is that relative decrease in cost facilitated by the NRS is higher for more popular items, be it a local or an external object. For low $\phi^{\text{as-as}}$, there is no need to index external content as Figs.2(a) and 2(d) show. Since inter-AS routing is not costly, it is more favourable to search for the content in the network and fetch it from external ASes in case of failing to find it via FBS. On the other hand, in case of the NRS assistance, content will be fetched from an in-network cache with average hop distance being higher than $\phi^{\text{as-as}}$, e.g., for AS 1755, $\bar{h} = 4.53$ vs. $\phi^{\text{as-as}} = 3$.

For local content, fraction of improvement is consistent across different ASes and significantly lower than external items (for moderate and high $\phi^{\text{as-as}}$). Fraction of improvement is around only 2% for local hot content, whereas it may go up to 32% for external hot content depending on $\phi^{\text{as-as}}$. After a closer look to the change in cost of discovery and retrieval, we observe that local content benefits from the NRS in discovery phase significantly. More particularly, local content's discovery cost (figure not plotted) decreases by 88% for AS 3967, 85% for AS 1755, 93% for AS 3257, and 97% for AS 1239. The highest improvement of AS 1239 is due to its strongly connected structure compared to other ASes. In AS 1239, average node degree is 6.17 and therefore a content discovery message with scope 3 travels to many nodes resulting in high overhead. Hence, the NRS decreases the discovery cost drastically. Similarly, the second highest improvement occurs at AS 3257, which has the second highest average node degree. In short, improvement in discovery phase is a function of average node degree. As for retrieval cost, the NRS does not help decreasing retrieval cost for local content as we assumed $\rho = 0$ (we will also discuss the case when $\rho = 1$). Considering these two costs, for $s_k = 1$ Mb, retrieval cost dominates the total cost of content delivery. Hence, overall the improvement in utility is marginal.

Increasing $\phi^{\text{as-as}}$ has the same effect on all ASes: it becomes more appealing to store the information about external content in the NRS as the associated cost of retrieving these items is very high. As Figs.2(c) and 2(f) show, even for very popular items which are expected to be accessible via FBS, it is desirable to index their locators in the NRS.

Fig.3(a) shows the change in ΔU_k for AS 1755 and high $\phi^{\text{as-as}}$, $s_k = 1$ Mb, $\rho = 0$. Similar to the well-known relation between caching benefit and the content popularity, indexing gain ΔU_k is a monotonously decreasing function of popularity. As we see in the figure, while the same conclusion holds for popular local content, their utility ΔU_k is lower than external content for high $\phi^{\text{as-as}}$. This result is somewhat interesting as we expect popular items being easily accessible in the network, hence eliminating the need to keep track of their locators. However, the NRS not only decreases the costly FBS for content discovery but also avoids duplicate transmission of the relatively large content chunks. For the former, the message size is small, however its coverage can be extensive de-



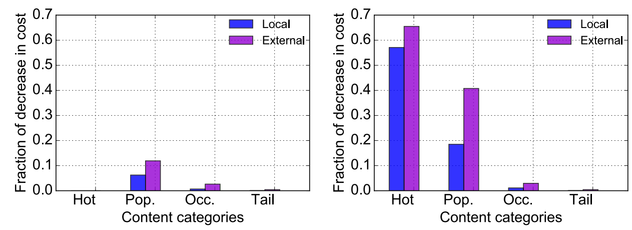
(a) ΔU_k for AS 1755. (b) Content size=100 Kb. (c) Content size= 1 Mb.

Figure 3: (a) Change in indexing gain with decreasing popularity. (b, c) Breakdown of cost saving contributed by discovery and retrieval for external content across different ASes for high $\phi^{\text{as-as}}$ and under two content size values. Numbers in the legend of (c) represent the mean values for each empirical CDF curve. Please see the legend of (c) for line colors and corresponding setting in (b).

pending on the scope parameter of FBS and the network connectivity. For the latter, despite entailing a restricted number of transmissions en-route to the requester, the message size is large and even a few redundant transmissions may result in significant bandwidth waste. Hence, indexing hot content becomes favourable. Figs.2(b), 2(c), 2(e), and 2(f) suggest that *significant improvements are observed only for hot and popular external content for this setting, whose merged set corresponds to a tiny fraction of the whole content catalogue, i.e., 1.1% including the local content.*

Figs.3(b) and 3(c) illustrate the saving facilitated by the NRS for *external* content due to both saving in discovery cost and saving in retrieval cost for high $\phi^{\text{as-as}}$ across the AS topologies. For each content, we calculate the fraction of cost saving due to the saving in the content discovery as $\frac{\phi_k(x_k=0)-\phi_k(x_k=1)}{\Delta U_k}$ and saving due to the decrease in the retrieval cost as $\frac{\beta_k(x_k=0)-\beta_k(x_k=1)}{\Delta U_k}$. For mean $s_k = 100$ Kb, cost saving is both due to the comparable savings in discovery and retrieval cost. More particularly, saving in discovery cost contributes to 17% of saving in total cost for AS 3259 with $N = 79$ routers, 16% for AS 1755 with $N = 87$ routers, 29% for AS 3257 with $N = 161$ routers, and 46% for AS 1239 with $N = 315$ routers. The high improvement in AS 1239 compared to other ASes is due to its higher connectivity resulting in too much FBS overhead. For such a network, the NRS prevents costly search for content discovery. Similarly, discovery cost of local content decreases by 97% for AS 1239, 93% for AS 3257, and 85% for AS 1755. Unsurprisingly, when mean content size is $s_k = 1$ Mb, main benefit of the NRS lies in its power to fetch the nearest content copy in off-path caches and decrease the average hop between the requester and the provider. For instance, AS 1755 gains the most for content retrieval, i.e., on the average 98% of the cost savings by the NRS is due to the lower cost of content retrieval.

Setting $\rho = 1$, we expect that ΔU_k is dominated by the saving in content retrieval, especially for high mean content size, i.e., s_k/l^{req}. However, when availability of object is very low in the network, despite high value of $\rho = 1$, redundancy in content retrieval will also be very low. Hence, we expect the effect of ρ to be more visible for networks where cache capacity is large so that expected number of copies of some content in FBS neighborhood is more than one. For such cases, we expect NRS-based operation to be more efficient in networks where inter-node synchronization is slower or weak, and thereby exhibiting high ρ values. Fig.4 plots the fraction of decrease in cost under both $\rho = 0$ and $\rho = 1$ for *large cache regime*, e.g., network's aggregate cache capacity is large enough to store 10% of the content catalogue. Fig.4(a) suggests that there is no need to index hot content when the network is perfectly synchronized and only one content copy is routed to the requester. Hot content can be discovered in one of the neighbors' caches. However, when $\rho = 1$, the NRS enables approximately 55% saving in the cost for local hot content and 65% saving in the cost for external hot content by preventing redundancy in content retrieval. Similarly, for popular content, cost savings are around 40% for external content and 19% for local content vs. 12% and 6% when $\rho = 0$. Our result also highlights the need to fine-tune the flooding-based content discovery and need for communication among nodes to decrease its wastefulness. Applying scope optimization [23] may help or more conservative content discovery schemes can be applied cautiously.



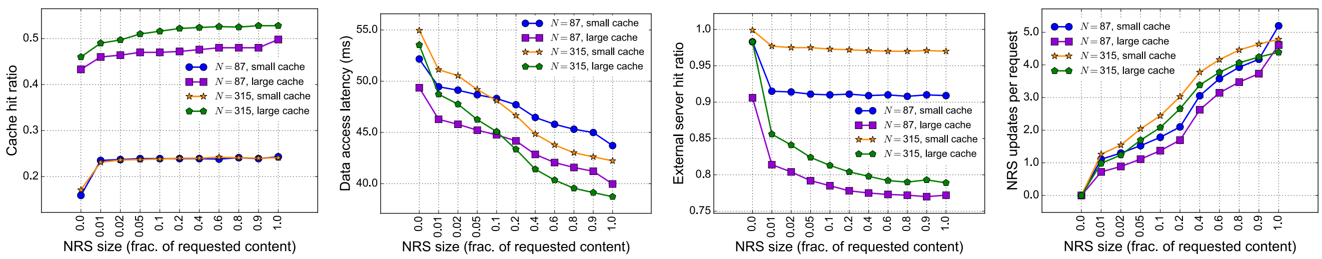
(a) $\rho = 0$. (b) $\rho = 1$.

Figure 4: Impact of ρ on indexing gain for *large cache regime*, AS 1755, $s_k = 1$ Mb, and moderate $\phi^{\text{as-as}}$.

5.2 Impact of NRS Size (ω)

In the previous section, we based our analysis on our utility function, which has some shortcomings to capture the real world complexities, e.g., on-path cache hits. Now, we relax our assumptions and evaluate the performance of NRS-based ICN with increasing NRS size, ω in Eq.(9), via simulations. We find the optimal NRS indexing scheme at the beginning of each simulation run and apply NRS-based content delivery based on the NRS setting.

We set the size of content catalogue to 4×10^4 and generate 10 requests per second in the network summing up to total 5×10^5 requests. For the first 10^5 requests, we do not record the statistics to account for warm-up period. The resulting traffic volume for external content is approximately half of the total user requests. Each router manages its cache space using LRU. Regarding the latency of each link, we use the values suggested by ICARUS simulator; 2 ms for intra-AS and 34 ms for inter-AS links based on [18, 26].



(a) Cache byte hit ratio. (b) Data access latency. (c) External server hit ratio. (d) Number of NRS updates.

Figure 5: Impact of NRS size on the network performance for $K = 4 \times 10^4$, $s_k = 1$ Mb, and high $\phi^{\text{as-as}}$.

Fig.5 depicts the change in cache byte hit ratio, external content's server hit ratio to reflect the savings in inter-AS traffic, data access latency, and number of NRS update messages per request. We observe in Fig.5(a) that for small cache capacity, an NRS which indexes only 1% of content catalogue improves cache byte hit ratio from around 16–17% to 23% for these ASes. While there is further improvement in cache hit ratio with increasing NRS size, the benefit is only marginal: increases from 23% to 24%. This result supports our observation in Sec.5.1, where we observe substantial decrease in cost only for hot and popular content. For larger cache capacity, we observe the highest performance improvement (from 43% to 49% for $N = 315$, and from 41% to 46% for $N = 87$) when we deploy the NRS to the system with $\omega = 0.01$. However, in contrast to an almost zero benefit for small cache capacity for $\omega > 0.01$, larger NRS further increases cache byte hit ratio. This is due to higher availability of content in the network for larger cache capacity. The NRS can increase the cache byte hit ratio only if the content exists in the network. *Hence, it is fair to state that content availability provides a bound on the benefits the NRS can provide and the impact of the NRS is more visible under larger cache regime.*

In contrast to diminishing gain in the cache hit, Fig. 5(b) shows that decrease in data access latency persists with increasing NRS size. We attribute this effect to the removal of the first step of content delivery, i.e., FBS-based content discovery. The NRS informs the requesting router about the existence (most likely the lack) of the content in the network so that router skips the FBS and directly routes its request to the origin server. Note that in either case, i.e, existence of the NRS and lack of it, the content that is unlikely to be in the network, e.g., far tail content, is retrieved from the origin server. Hence, we do not observe significant increase in the cache hit ratio for $\omega > 0.05$. However, the NRS assistance helps nodes to save the time wasted in the FBS, which results in lower data access latency in Fig.5(b).

Similarly, Fig.5(c) shows that fraction of user requests served from external networks decreases with increasing NRS size. The server hit ratio decrease between the two points corresponds to the (cost) savings in the inter-AS traffic. For example, for AS 1239 ($N = 315$), the saving is 2% only for small cache and 13% for large cache capacity when NRS indexes 1% of the requested content compared to the case of no NRS. For AS 1755, cost savings are 6% and 11%, respectively. The lower saving of AS 1239 for small cache capacity is due to the smaller per router cache capacity in AS 1239 which lets only 1-2 objects in the cache as compared to 4-5 objects in AS 1755. Moreover, for both cases, inter-AS traffic can further be saved if the caches implement some cache management policy other than LRU which does not differentiate between local and external content. As for intra-AS traffic (figure not plotted), the effect of the NRS is more intertwined: while the NRS reduces intra-AS traffic by avoiding the FBS-triggered duplicate content transmissions, it also increases intra-AS traffic by reducing inter-AS traffic. Overall, our results show that both effects cancel each other, resulting in almost no difference in intra-AS link loads.

Clearly, a larger NRS requires more traffic between the in-network caches and the NRS. With increasing NRS size, the control overhead increases linearly as Fig.5(d) shows. In fact, control updates for an incoming request is a linear function of the number of routers on the content path. However, as the NRS tolerates certain levels of distortion, i.e., false alarm and false negative, the update per session is lower than content hop count: approximately 1 message for $\omega = 0.01$ and 4 messages for $\omega = 1$ under our setting $\varepsilon^1 = \varepsilon^0 = 0.1$.

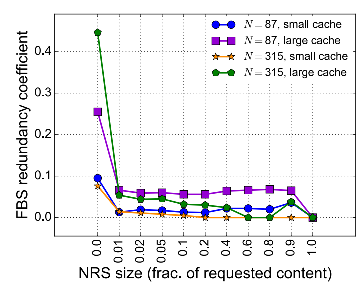


Figure 6: Change of ρ with increasing NRS size.

Finally, Fig.6 depicts the change in ρ, FBS redundancy coefficient, with increasing NRS size. We calculate redundancy as $\rho = (L - l_{nc})/l_{nc}$ where L stands for the total number of transmissions (i.e., links traversed) of the content as a response to FBS messages and l_{nc} is the number of links traversed by the nearest copy of the content towards the requesting router. Note that this current definition of redundancy reflects the actual waste of bandwidth due to redundant transmissions as opposed to our definition of redundancy in Section 3.3, which we preferred for the sake of simplicity. In agreement with our results in Fig.4, overhead can be very high for an ICN with large cache capacity in the absence of NRS, e.g. 25-40% depending on the topology. With increasing NRS size, frequency of FBS decreases, which explains the decrease in ρ in Fig.4. Finally, when all items are indexed, i.e., $\omega = 1$, FBS never takes place as the NRS provides information about the actual location of each content and FBS redundancy equals to zero.

5.3 Discussion

In our models, we have not accounted for temporal [22] or spatial locality [7] within the AS domain, which increases the chances of finding the content in the neighborhood of the requesting router. Higher locality leads to higher cache hit ratio, but at the same time may result in higher ρ. Therefore, we expect the NRS to provide more benefits under higher locality. However, a careful analysis using representative traffic traces is needed to show the validity of our claim. Such an analysis will also help understanding what fraction of the requests is for local content. In our paper, we have not considered different content configurations, e.g., a larger AS may have a higher fraction of local content whereas a higher fraction of traffic will be for external content for a smaller AS. We reserve an in-depth analysis on the effect of content requests and the impact of the indexing decision of an AS on other ASes as future work.

The NRS provides performance improvement in two ways; in the discovery and in the retrieval, or in terms of time and the bandwidth. First, if the requested content is not in the network, the NRS avoids the waste of time for the exploration of a content copy which would eventually fail to find the requested content. Second, if the content is in the network, the NRS ensures that only the nearest copy of the content is retrieved without any redundant transmission. Our analysis suggest that the NRS benefits is quite significant especially for the latter as the size of a packet carrying the content is larger compared to a content discovery message. However, our model assumes a network applying scoped-flooding for content discovery, which is not necessarily the only way for content discovery. For discovery schemes which are more parsimonious in spreading content discovery messages, the impact of the NRS in bandwidth saving may be lower. However, the impact in terms of time saving may then be higher as a more parsimonious scheme with less parallelism in search needs longer time for content discovery.

6. RELATED WORK

As scalability of name resolution has been a common concern in the ICN community, there is a considerable number of related work, e.g., a new protocol layer to overcome the challenges of name-based delivery [17], analysis of ICN naming architectures [13], or a proposal to improve the naming scalability of NDN [1]. However, we will keep our discussion to the most relevant ones as follows.

Mainly driven by the scalability concerns, Azimdoost et al. [3] propose that nodes update the NRS only periodically. Using rate distortion theory, authors quantify the required rate of NRS updates from the caches for each content to guarantee accuracy below a certain level of distortion, e.g., false positives. Based on the determined update rate, [3] calculates the expected operation cost, i.e., downloading the content and updating the NRS. While we adopt the same approach, as opposed to a constant content download cost in [3], we consider the expected location of the requested content in the network to reflect the fact that content download cost is a function of how many routers are involved in the process, e.g., bandwidth usage, processing cost, or transmission delay at each hop. Second, our work differs from [3] in that rather than indexing all items, we identify which items the NRS should index for the minimum content delivery cost given a constraint on the number of objects to index.

There are also decentralized content indexing proposals where each router announces its cached contents to its neighbors so that each router can index locally the content in its neighborhood. Similar to the centralized NRS, ensuring the consistency of the announced information is challenging due to volatile content in the caches. Wang et al. [24] cope with this challenge by slowing down the cache replacement and advertising contents of each router in a limited scope. Each cache makes commitments on the duration of keeping each content in cache such that announced cache information remains valid till the next content advertisement (e.g., similar to TTL caches [12]). Similarly, SCAN [15] extends the lifetime of the content in the cache till the next advertisement period by storing these items in a buffer so-called protected storage. Next, based on the duration of the commitment period, routers determine the advertisement scope, e.g., within three hops, for their content. Since advertising the list of all content is costly, both [15] and [24] use bloom filters — which are probabilistic data structures ideal for representing a set with some probability of false positives. Our position is very similar to [15, 24] in that we advocate that content discovery can be improved at the control layer with the NRS which does not necessarily know everything but knows the most useful pieces of information. However, different than these works, we elaborate on which pieces of information the NRS should know.

Chiocchetti et al. [6], motivated by questions similar to ours, devise a hybrid scheme which exploits the existing knowledge in the FIBs and explores the node's neighbors via flooding in the lack of such information. However, our approach differs from [6] in that we derive the optimal NRS setting by first formulating an optimization problem that minimizes the total cost of content delivery under a particular NRS size constraint. The solution to our problem gives us the optimal NRS configuration, and next using real AS topologies we identify for which items exploration would be wasteful and exploitation should rather be preferred considering the locality of content and the associated cost of inter-AS routing for external content. Our results differ from those in [6] for networks where exploration may trigger multiple content copies to be routed back to the requester due to the lack of synchronization among nodes. Hence, our results suggest that the NRS can bring the highest cost saving for the most popular items by avoiding wasteful exploration, in contrast to [6] which proposes exploration for the popular content.

Lastly, [9] provides key design requirements of an NRS, e.g., failure resilience, in addition to a comparison of a standalone NRS and name-based routing. To favour scalability of the NRS, our proposal consciously relaxes the requirement of *resolution guarantee* listed in [9], which refers to the desirable property that an NRS must be capable of locating every content in the caches regardless of the content's properties. The NRS which indexes only a fraction of the content catalogue postpones the resolution of the un-indexed content to the content discovery scheme, scoped-flooding in our case, which may not ensure resolution.

7. CONCLUSION

In this article, we have modelled and quantified the related impact of a Name Resolution Server (NRS) on each content's delivery cost based on its popularity, size, availability, and its locality, i.e., local or external content. Then, we have identified the content objects whose delivery cost substantially decreases in the existence of the NRS by solving our optimization problem which minimizes the expected cost of content delivery within an AS given a size constraint for the NRS. Our results suggest that an AS can increase its cache hit ratio, decrease its inter-AS traffic and data access latency with the help of an NRS which indexes only a small fraction of the requested content set. This small fraction corresponds to the frequently requested content and especially external content among these popular items, as inter-AS traffic is costly. Potential future directions are analysis of our proposal on a more realistic setting, e.g., Internet AS topology and realistic traffic, and how the indexing decision of an AS affects the other ASes.

8. ACKNOWLEDGMENTS

Arjuna Sathiaseelan was funded by the EU H2020 RIFE project - grant No.644663 (Action full title: architectuRe for an Internet For Everybody, Action Acronym: RIFE).

9. REFERENCES

[1] A. Afanasyev, C. Yi, L. Wang, B. Zhang, and Zhang. SNAMP: Secure namespace mapping to scale NDN forwarding. In *IEEE Global Internet Symposium (GI 2015)*, 2015.

[2] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. A survey of ICN. *IEEE Comms. Magazine*, 50(7):26–36, 2012.

[3] B. Azimdoost, C. Westphal, and H. R. Sadjadpour. The price of updating the control plane in ICNs. *CoRR*, abs/1406.1284, 2014.

[4] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and B. Mathieu. A survey of naming and routing in ICN. *IEEE Comms. Mag.*, 50(12):44–53, 2012.

[5] H. Che, Y. Tung, and Z. Wang. Hierarchical web caching systems: Modeling, design and experimental results. *IEEE Journal on Selected Areas in Comms. (JSAC)*, 20(7):1305–14, 2002.

[6] R. Chiocchetti, D. Rossi, G. Rossini, G. Carofiglio, and D. Perino. Exploit the known or explore the unknown?: Hamlet-like doubts in ICN. In *ACM ICN*, 2012.

[7] A. Dabirmoghaddam, M. M. Barijough, and J. Garcia-Luna-Aceves. Understanding optimal caching and opportunistic caching at the edge of ICN. In *ACM ICN*, 2014.

[8] C. Dannewitz, M. D'Ambrosio, and V. Vercellone. Hierarchical DHT-based name resolution for ICNs. *Comp. Comms.*, pages 736–49, 2013.

[9] L. Dong, C. Westphal, G. Wang, and J. Wang. Requirements of Name Resolution Service in ICN, *work in progress*. Internet-Draft draft-dong-icnrg-nrs-requirement-00, IETF Secretariat, July 2016. http://www.ietf.org/internet-drafts/draft-dong-icnrg-nrs-requirement-00.txt.

[10] M. Draxler and H. Karl. Efficiency of on-path and off-path caching strategies in ICN. In *IEEE GreenCom*, 2012.

[11] N. Fotiou, D. Trossen, and G. C. Polyzos. Illustrating a publish-subscribe Internet architecture. *Telecommunication Systems*, 51(4):233–245, 2012.

[12] J. Jung, A. W. Berger, and H. Balakrishnan. Modeling TTL-based Internet caches. In *IEEE INFOCOM*, pages 417–426, 2003.

[13] K. V. Katsaros, X. Vasilakos, T. Okwii, G. Xylomenos, G. Pavlou, and G. C. Polyzos. On the inter-domain scalability of route-by-name ICN architectures. In *IFIP Networking*. 2015.

[14] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. In *ACM SIGCOMM Comp. Comm. Review*, volume 37, pages 181–92, 2007.

[15] M. Lee, J. Song, K. Cho, S. Pack, T. T. Kwon, J. Kangasharju, and Y. Choi. Content discovery for information-centric networking. *Computer Networks*, 83:1 – 14, 2015.

[16] K. Mokhtarian and H.-A. Jacobsen. Coordinated caching in planet-scale CDNs: Analysis of feasibility and benefits. In *IEEE INFOCOM*, 2016.

[17] I. Psaras, K. V. Katsaros, L. Saino, and G. Pavlou. LIRA: A location independent routing layer based on source-provided ephemeral names. *arXiv:1509.05589*, 2015.

[18] J. Rajahalme, M. Särelä, K. Visala, and J. Riihijärvi. On name-based inter-domain routing. *Comp. Networks*, 55(4):975–86, 2011.

[19] L. Saino, I. Psaras, and G. Pavlou. Icarus: a caching simulator for ICN. In *SIMUTOOLS*, 2014.

[20] A. Sathiaseelan, L. Wang, A. Aucinas, G. Tyson, and J. Crowcroft. SCANDEX: Service centric networking for challenged decentralised networks. In *Workshop on Do-it-yourself Networking: an Interdisciplinary Approach*, 2015.

[21] T. Song, H. Yuan, P. Crowley, and B. Zhang. Scalable name-based packet forwarding: From millions to billions. In *ACM ICN*, 2015.

[22] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi, and S. Niccolini. Temporal locality in today's content caching: why it matters and how to model it. *ACM SIGCOMM Comp. Comm. Review*, 43(5), 2013.

[23] L. Wang, S. Bayhan, J. Ott, J. Kangasharju, A. Sathiaseelan, and J. Crowcroft. Pro-diluvian: Understanding scoped-flooding for content discovery in ICN. In *ACM ICN*, 2015.

[24] Y. Wang, K. Lee, B. Venkataraman, R. L. Shamanna, I. Rhee, and S. Yang. Advertising cached contents in the control plane: Necessity and feasibility. In *IEEE INFOCOM NOM*, 2012.

[25] W. K. Wong, L. Wang, and J. Kangasharju. Neighborhood search and admission control in cooperative caching networks. In *IEEE GLOBECOM*, 2012.

[26] B. Zhang, T. Ng, A. Nandi, R. Riedi, P. Druschel, and G. Wang. Measurement based analysis, modeling, and synthesis of the Internet delay space. In *ACM SIGCOMM conference on Internet measurement (IMC)*, pages 85–98, 2006.

[27] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang. Named data networking. *ACM SIGCOMM Computer Communication Review (CCR)*, 44(3):66–73, 2014.