Bloom Filter based Inter-domain Name Resolution: A Feasibility Study

Konstantinos V. Katsaros, Wei Koong Chai, George Pavlou
Dept. of Electrical & Electronic Engineering, University College London
WC1E 7JE, Torrington Place, London, UK
{k.katsaros, w.chai, g.pavlou}@ucl.ac.uk

ABSTRACT

The enormous size of the information space raises significant concerns regarding the scalability of name resolution in Information-Centric Networks, especially at a global scale. Recently, the use of Bloom filters has been proposed as a means to achieve a more compact representation of name resolution state. However, little attention has been paid to the expected performance or even the feasibility of such an approach. In this paper, we aim to fill this gap, presenting a feasibility study and performance analysis of Bloom filter based *route-by-name* inter-domain name resolution schemes. We propose a methodology for assessing the memory and processing resource requirements of the considered schemes and apply it on top of the inter-domain topology. Our investigation reveals that the skewed distribution of state across the inter-network results in a hard to balance trade-off between memory and processing resource requirements. We show that hardly any Bloom filter configuration i.e., size and bits-per-element, is able to reduce both types of resource requirements for all Autonomous Systems (ASes) in the network, while lowering resource requirements at one area of the inter-domain topology inflates resource requirements at another. Detailed simulations further show that the direct connection of multiple stub networks to tier-1 ASes, results in a dramatic increase of false positives, questioning the reliability of a BF-based inter-domain name resolution scheme.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Distributed networks*

Keywords

ICN; scalability; false positive; DONA; CURLING.

1. INTRODUCTION

The Information-Centric Networking (ICN) paradigm focuses on the dissemination and retrieval of information rather that the pairwise communication between end hosts. In ICN, information is organized into named Information Objects (IOs) and a name resolution process is responsible for guiding forwarding decisions based on the names of the requested IOs. This mode of operation enables a series of desirable features such as in-network caching, multicast and mobility support, that have motivated substantial research efforts for the adoption of ICN in the context of the Internet (e.g., see [25] and references therein).

The applicability of the ICN paradigm at the Internet-scale heavily depends on the scalability of the name resolution system (NRS). At this scale, the NRS is required to support routing and forwarding for the entire IO population in the Internet, whose sheer size raises significant feasibility concerns. The size of the indexable Web has been estimated in the order of 10^{12} web pages [12], while the advent of the Internet of Things / Internet of Everything, yields an expectation for even up to 50 billion interconnected devices [8]. These estimations have led to different projections for the size of the IO namespace, ranging from 10^{13} [18] to 10^{17} [6]. Obviously the actual impact on the NRS heavily depends on the granularity of the selected naming scheme, which is yet to reach consensus in the ICN research community[1].

Currently on-going research efforts aim to address this scalability concern with the use of Bloom filters (BFs) [3], under the premise that with a *small* cost of possible but rare incorrect content request resolution, the entire system gains *much* in terms of scalability and performance. In such schemes, IO names are inserted into fixed size BFs which are maintained throughout the inter-domain topology so as to denote the availability of the corresponding content in neighboring Autonomous Systems (ASes). Using BFs results in the compact representation of content availability. It is also expected to lower network traffic and lookup overheads i.e., only a single operation is required to check IO availability in a BF, as opposed to a search procedure through the entire corresponding set of IO names. The cost of these benefits is the possibility of having false positives, in which BF membership queries erroneously report content availability leading to a failing name resolution process.

So far, the use of BFs for the support of inter-domain name resolution schemes in ICN has only been investigated at a coarse level, largely neglecting the details of the BF configuration, namely, the size and *bits-per-element* ratio of BFs, as well as their impact on both the performance and the resource requirements of the resulting NRS. In this

[1] In this paper, we take a conservative approach, assuming a total population of $S = 10^{13}$ IO names.

ICN'15, September 30–October 2, 2015, San Francisco, CA, USA.
© 2015 ACM. ISBN 978-1-4503-3855-4/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2810156.2810175.

paper, we aim at filling this gap by delving into the details of the complex inter-dependent impact factors contributing to the tradeoffs of the resolution system utilizing BFs. Our study seeks to verify if the exploitation of BFs can actually fulfill its promise on solving the scalability issue of NRS in ICN. We provide quantitative and illustrative answers to questions on how *small* exactly is the price we need to pay and how *much* it is the individual ASes and the system as a whole stand to gain.

Specifically, we first provide a model enabling the assessment of the suitability of a BF configuration in terms of resource requirements, paying particular attention to the role of the distribution of name resolution states across the inter-domain topology. Focusing on the case of a *route-by-name* name resolution model, e.g., [19, 5], we study this interplay, based on the AS-level Internet graph inferred by the CAIDA BGP traces [1]. Our study reveals that the need for a fixed Internet-wide BF configuration results in considerably diverse resource requirements throughout the inter-domain topology. Namely, large BFs result in large volumes of underutilized computational resources for stub ASes, which constitute the vast majority of ASes in the Internet, since in these domains, BFs tend to be sparse due to the low number of the registered IOs. On the other hand, smaller BF sizes substantially increase resource requirements at the higher tiers of the inter-domain hierarchy for two reasons: (i) smaller BF sizes result in a larger number of BFs to be looked-up for each name resolution request, (ii) for a larger number of BFs, a higher number of BF bits-per-element is required to avoid inflating the system-wide false positive ratio. Building on the introduced model and the CAIDA trace set, we conduct a realistic empirical study on the selection of an appropriate BF configuration and discuss the respective costs, gains and tradeoff space.

Furthermore, we complement our investigation with detailed packet-level simulations to shed light on a set of important, but rather complex to model, aspects of the envisioned BF-based NRSes. Namely, we assess the impact of false positives on the overall NRS performance, across the Internet. Moreover, we assess the effectiveness of applying a *bin-packing* algorithm for the consolidation (i.e., merging) of BFs during state propagation. Finally, we assess the processing overheads associated with the volume of name resolution requests across then network.

The findings of this paper are summarized as follows:

- There is no BF configuration that can lower both memory and processing resource requirements for *all* AS in the inter-domain topology, compared to a BF-agnostic scheme.
- There is no BF configuration that can lower the volume of one type of resources (i.e., memory or processing) without inflating the requirements on the other, either locally or in some other area of the inter-domain topology.
- The direct connection of multiple stub networks to tier-1 ASes inflates the number of BFs maintained by the latter, dramatically increasing the rate of false positives.

The remainder of this paper is organised as follows. In Section 2 we provide a detailed description of the current state-of-the-art in inter-domain name resolution in the context of ICN, motivating the need for the feasibility study presented in this work. Next, in Section 3 we present the Bloom filter configuration parameters and their impact on the resulting resource requirements, revealing important configuration trade-offs. In Section 4 we investigate the interplay between the state distribution across the inter-domain hierarchy and the configuration of BFs, quantifying its impact on the resulting memory and processing resource requirements across the topology. In Section 5, we further employ simulations to shed some light on more complicated aspects of the considered NRSes.

2. BACKGROUND

Research efforts on the support of inter-domain name resolution, largely fall into two basic categories. *Lookup-by-name* approaches decouple name resolution from data forwarding by realizing name resolution as a directory service [6, 17, 15]. In most cases, such solutions rely on Distributed Hash Tables (DHTs) in order to perfectly load balance the overheads imposed from the enormous size of the IO namespace, i.e., memory requirements for name resolution state, processing overheads for lookup operations. However, it has been shown that such approaches present inefficient routing due to stretched name resolution paths and inter-domain routing policy violations [11, 17].

Route-by-name approaches, on the other hand, offer shortest path, BGP-compliant routing. In the classic route-by-name approach, DONA [19], each IO is associated with a *principal* that can be considered as its owner. The DONA design involves an overlay of *Resolution Handlers* (RHs) with at least one logical RH placed at each AS. The role of RHs is to register and maintain name resolution state, as well as to guide the propagation of name resolution requests until they are resolved. To this end, RHs interconnect following the hierarchical business relationship interconnection of their ASes, forming a corresponding RH hierarchy. The RH hierarchy is further enhanced with peering links, i.e., RHs of peering domains are also linked in the RH overlay[2].

Principals issue REGISTER messages towards their local RH(s) to advertise their IO's to the network. A local RH propagates a REGISTER message upwards to its providers in the inter-AS hierarchy and to RHs at peering ASes, thus setting up the name resolution state throughout the network. The propagation of REGISTER messages terminates at the top-most AS level, i.e., at tier-1 RHs (see Section 4). ASes not willing to transit name resolution requests and/or data do not propagate REGISTER messages received over peering links. Since tier-1 ASes all peer with each other, each tier-1 AS will be aware of all IOs in the network. The resolution state at each RH is in the form of <`IO name, next hop RH`> pairs, i.e., mappings between advertised IO names and a pointer to the previous RH in the corresponding REGISTER propagation path.

Clients (i.e., end hosts) issue name resolution requests in the form of FIND messages submitted to their local RH(s). FIND messages are also propagated upwards via provider-customer links in the domain hierarchy according to the inter-domain routing policies, but not over peering links. In the worst case, a FIND message has to reach a tier-1 AS

[2]Since peering links introduce cycles in the inter-domain network graph, the topology is not strictly hierarchical.

in order to locate an IO, or determine that no such IO exists. Upon a name match with an RH entry, FIND messages follow the reverse registration path to reach the local RH of the appropriate principal, which triggers the data transfer.

CURLING [5] follows a similar approach, also adapting to the inter-domain topology structure for name registration and resolution. However, in CURLING, *Content Resolution Servers* (CRS)[3] propagate registration and resolution requests only to their provider ASes. In effect, both REGISTER and FIND messages follow a subset of the underlying routing relationships, as they do not cross peering links. As a result, when a FIND request reaches a tier-1 AS, it will have to be broadcasted to all other tier-1 ASes to guarantee resolution. CURLING however allows optimizing the data paths to allow the utilization of peering links in the delivery of the content itself.

The compliance of name resolution paths to valid BGP shortest path, in DONA and CURLING, comes at the cost of non-negligible replication of name resolution states across the Internet. This results in extensive resource requirements to be satisfied by the participating ASes. As shown in our previous work [18], for a population of $S = 10^{13}$ IOs, the memory footprint of name resolution state reaches 420 TB. This corresponds to the capacity of an entire small to medium scale data center, if state is to be maintained in RAM, e.g., the DONA scheme would require more than 26,000 16GB RAM servers for tier-1 ASes, on average.

These enormous resource requirements have triggered alternative approaches, focusing on a more space-efficient representation of name resolution state. To this end, the use of Bloom filters [3] has been proposed in [21, 14]. Liu et *al.* proposed the use of Bloom filters for the representation of name resolution state [21]. Though promising a significant reduction of the name resolution state size, the presented scheme employs DHTs in the name resolution process, raising the aforementioned concerns on routing quality. Hong et *al.* present an alternative approach, in which name resolution state is disseminated following the inter-domain hierarchy, similarly to DONA and CURLING, thus promising efficient name resolution paths. All NRS nodes in the inter-domain topology use the same set of hash functions to add the names of IOs of local content providers into globally fixed size BFs. Each NRS node maintains a name lookup table that contains the mappings between the IO identifiers registered by CPs in the local domain and the corresponding network locators, i.e., the information required to reach the content. Additionally, an NRS node maintains a set of Bloom filters describing the availability of IOs in customer and (possibly) peering domains[4]. Each NRS node further creates a BF for the locally registered state and forwards it to each provider and (possibly) peering domain along with the BFs received from customer domains. The propagated BF(s) constitute(s) the union[5] of all BFs received from customer domains (along with the entries for locally registered content). Following [14], we assume BFs being periodically propagated by participating nodes, so that the overhead of updating the corresponding name resolution state is controlled. The propagation of BFs and name resolution requests follows the underlying topology, in a fashion similar to DONA and/or CURLING. We thus term the corresponding BF-enabled schemes as DONA-BF and CURLING-BF, depending on whether BFs are propagated over peering links.

Though BFs appear as a promising solution towards the alleviation of the discussed state size related overheads, we still lack a good understanding of the behaviour of a BF-based NRS. The existence of BF false positives calls for a detailed investigation of the configuration of the BFs, i.e., the size and *bits-per-element*, so as to gain insights into the overall performance of a BF-enabled NRS. This includes both the failure of the system to resolve names and the resulting resource requirements, which are affected by the distribution of IOs across the Internet i.e., the volume of name resolution state to be represented. In the following section, we provided a detailed description of a proposed framework for the evaluation of the considered NRSs.

3. IMPACT OF BF CONFIGURATION

In this section, we discuss the parameters that are available to configure BFs with, which we can use as "tuning knobs", and their related impacts and tradeoffs to the system performance and resource requirements.

3.1 Bloom Filter Preliminaries

Bloom filters (BFs) are probabilistic representations of sets [3]. A BF is represented as an array of m bits initially set to 0. An element is inserted into a BF with the help of k hash functions, i.e., each function maps the hash of the inserted element to a position in the bit array which is set to 1. To check whether an element belongs to a set, the k bit array positions indicated by the hash functions are checked. If an element belongs to a BF, then all k positions are set to one and the item is found with probability 1. BFs present a non-zero *false positive ratio* (R), i.e., a BF may falsely report the presence of an element. Given the length of the BF (m), the number of inserted elements (n), and the number of hash functions (k), R can be calculated as follows:

$$R \approx (1 - e^{-kn/m})^k \tag{1}$$

Using the optimal number of hash functions $k = \frac{m}{n} ln2$, we can calculate the bits-per-element ratio for a certain upper limit of R (defined as R_{max}), i.e.,

$$\frac{m}{n} = -\frac{lnR_{max}}{(ln2)^2} \tag{2}$$

Based on Equation 2, we can calculate the maximum number of items that can be inserted in a BF without exceeding R_{max}, for a given value of m, namely the *capacity* (C_{BF}) of the BF. Table 1 summarizes the notation used in this paper.

3.2 Configuring BFs

The non-zero false positive ratio of BFs has obviously an impact on the expected correctness of the name resolution process, i.e., false positives would lead to name resolution requests forwarded towards areas of the network where the corresponding content does not exist. Consequently, the design of a global scale NRS should aim at imposing an upper

[3]For simplicity, we will use the term RH for both DONA and CURLING, as RHs and CRSs offer similar functionality with respect to the aspects investigated.

[4]Though not explicitly described in [14], it is assumed that BFs are maintained per corresponding neighbor, so that a positive membership query can yield the appropriate forwarding information.

[5]Achieved by simply applying the bit-wise OR operation.

Table 1: Notation

Symbol	Definition
m	Size of BF (bits)
k	Number of hash functions
n	Number of elements inserted in a BF
R	False positive ratio
R_{max}	Maximum allowed R per BF
R_{max}^{Node}	Maximum allowed R per node
C_{BF}	BF capacity (under R_{max})
S	Number of unique IOs in the network
s	Number of IO registrations at a node
s_l	Number of IO registrations in s originating from local content providers
F	Number of BFs maintained by a node
M	State size maintained at a node (bytes)
LO	Number of BF lookups performed at a node, per single resolution request
RL	Number of resolution requests received by a node
b_{NonBF}	Byte overhead per plain registration
b_{BF}	Byte overhead per BF
p	Probability an IO is registered at a node
P_{TP}	Probability a registered IO is found in the locally maintained BFs
P_P	Probability of a positive (multi-)BF lookup
P_N	Probability of a negative (multi-)BF lookup
N	Number of name resolution nodes
w	Resource wastage per node
W	Resource wastage across all N nodes

limit for this impact so as to guarantee a minimum control on the reliability of the NRS. To this end, in this paper, we consider such a limit on an NRS node level, i.e., we define a global upper limit for the false positive ratio experienced at any NRS node (denoted as R_{max}^{Node})[6].

The experienced R_{max}^{Node} depends on the number of BFs used to represent the overall state. The representation of s items, requires F BFs of capacity C_{BF},

$$F = \left\lceil \frac{s}{C_{BF}} \right\rceil \quad (3)$$

Equation 3 expresses the optimal case of a perfect assignment of IO names to BFs. In practice, the total number of BFs maintained by a NRS node depends on the union (or *merging*) operation applied by each node (see Section 2). In this work we consider this merging process to be enabled by a bin-packing algorithm in which BFs represent the bins to be filled with items in BFs received from customer ASes. All bins have capacity C_{BF}. Obviously, items in a received BF can only be merged into a new BF as a whole i.e., either all items in a received BF or none. Hence, the size of each item in the bin packing algorithm corresponds to the total number of items included in the received BF, which can be estimated or explicitly denoted by an item counter. As the merging process proceeds throughout the inter-domain hierarchy, received BFs are merged into new ones with increasingly more items. In effect the distribution of item sizes throughout the hierarchy varies. To the best of our knowledge, modelling the expected performance of bin-packing algorithms with a variable distribution of item sizes constitutes a research challenge on its own, and it is beyond the scope of this paper. We revisit the impact of this simplification in our model in Section 5.

[6]Note that R_{max}^{Node} does not represent the overall probability of a name resolution request failing due to a false positive. We revisit this issue in Section 5 where we assess this probability with simulations.

Under these assumptions, R_{max}^{Node} is given by Equation 4,

$$R_{max}^{Node} = 1 - (1 - R_{max})^F \quad (4)$$

It follows then that in order to guarantee R_{max}^{Node}, we need to select the C_{BF} and m values as shown in Equation 5. In the following, we term these two parameters as the *BF configuration*. In practice, and based on Equation 5, we can define different BF configurations by simply selecting different C_{BF} capacity values for a fixed R_{max}^{Node} constraint.

$$m = -C_{BF} \frac{ln(1 - (1 - R_{max}^{Node})^{\frac{1}{\left\lceil \frac{s}{C_{BF}} \right\rceil}})}{(ln2)^2} \quad (5)$$

Since a globally fixed BF configuration is required to guarantee inter-domain compatibility and the R_{max}^{Node} constraint applies to all nodes, the selected BF configuration should satisfy Equation 5 for all s values encountered across the Internet, even when the maximum possible number of elements is inserted in a BF. In our context, this is the case of tier-1 ASes, which need to maintain name resolution state even for the entire set of IOs in the Internet (S), in the case of DONA (see Section 2). Therefore, we consider the selection of a BF configuration among the ones satisfying Equation 5 for $s = S$.

3.3 BF Configuration Tradeoff

There are multiple candidate configurations satisfying the R_{max}^{Node} constraint and the exact selection requires the definition of the appropriate evaluation metrics. In the following section, we first define the main two evaluation metrics considered in this work, namely the memory and processing requirements and then discuss how the BF configuration impacts the system, whereby finding a Internet-wide optimal balance is not a trivial issue.

3.3.1 Memory requirements

As discussed in Section 1, the use of BFs is originally motivated by the need to constrain the overall state size to a scalable level (i.e., to reduce the requirements in memory resources to a level supportable by the current technologies). Here, we consider the case where all name resolution states are maintained in the main memory (i.e., RAM) so as to enable fast name resolution. In the context of our generic DONA-/CURLING-BF NRSs, the total memory requirements at a node hosting s registrations (out of a total population of S IOs), is given as follows:

$$M^x = b_{NonBF} \cdot s_l + b_{BF} \cdot F \quad (6)$$

where M^x is the total state size expressed in bytes, $x \in \{CURLING\text{-}BF, DONA\text{-}BF\}$, s_l is the number locally registered IO names i.e., $s_l < s$, $b_{NonBF} = 42$, is the byte overhead per local state entry[7] and $b_{BF} = (m + log_2 C_{BF})/8$ is

[7]b_{NonBF} corresponds to a 40 byte object identifier and a 2 byte pointer to the next RH i.e., a bitmap for the node's interfaces to the neighboring node[19].

the byte overhead of each BF, considering the size of the BF and the overhead of an item counter required to facilitate BF merging (see Section 3.2)[8]. Obviously for $x \in \{CURLING, DONA\}$, we have:

$$M^x = b_{NonBF} \cdot s \tag{7}$$

3.3.2 Processing overheads

Name resolution also incurs processing overheads related to the search of a IO name in the maintained name resolution state. The overall overhead depends on the number of BF lookup operations per received request, denoted here as the *lookup overhead* (LO). We approximate this (LO^x), for $x \in \{CURLING, DONA\}$, as follows:

$$LO^x = \alpha \cdot log(s) + c \tag{8}$$

where α denotes a positive real number[9]. When BFs are employed, the look-up overhead is affected by the total number of entries maintained by the nodes, which determines both the probability that an item is registered locally (i.e., $p = \frac{s}{S}$), and the number of BFs maintained, for a certain BF configuration. When $F > 1$ we assume that the maintained BFs are looked-up sequentially. In effect, the number of look-up operations is further affected by the false positive ratio i.e., a false positive results in no subsequent BF lookups. Hence, we estimate the expected look-up overhead as follows:

$$LO^x = F \cdot P_N^F + \frac{P_P}{P_N} \cdot \sum_{i=0}^{F} i P_N^i \tag{9}$$

for $x \in \{CURLING\text{-}BF, DONA\text{-}BF\}$, with P_P denoting the probability of a positive BF look-up response:

$$P_P = (1 - P_{TP}) R_{max}^{FP} + P_{TP} \tag{10}$$

where $P_{TP} = \frac{p}{F}$, is the probability that an existing entry is located in one of the maintained BFs and $P_N = 1 - P_P$.

The overall processing overhead at an NRS node also depends on the total number of name resolution requests received by the node, denoted here as the *resolution load* (RL). This overhead heavily depends on the structure of the inter-domain topology, as well as the effect of false positives. Given the complexity of the inter-domain topology we assess this aspect of processing overheads through simulations in Section 5.

3.3.3 Tuning BF Configuration

Inspecting Equations 6 and 9 shows that the overall resource requirements at each node of the considered BF-based NRS depend on the number of BFs maintained F, which in turn depends on the relation between the C_{BF} and the number of IO names, s. It follows that for a given value of s, the amount of memory and processing resources depends on the selected BF configuration.

[8]To simplify our analysis we neglect data structure overheads. We also do not consider interface pointer overheads, since BFs are maintained in a distinct set per neighbor.

[9]We base this approximation on the $O(logn)$ scalability properties of search algorithms, assuming that the size of the IO name space is sufficiently large to roughly approximate an asymptotic behaviour.

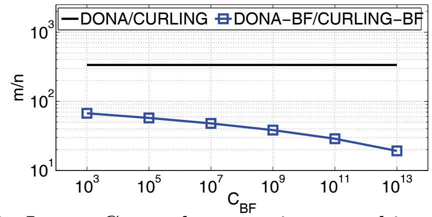


Figure 1: Lower C_{BF} values require more bits-per-element to satisfy the R_{max}^{Node} constraint. Both axes are in logarithmic scale.

On the one hand, selecting $C_{BF} >> s$ results in considerably sparse BFs. In such cases, the corresponding NRS nodes under-utilize the memory resources allocated for the maintenance of the resulting BF. Similarly, the propagation of such sparse BFs across the inter-domain hierarchy results in the waste of the corresponding bandwidth. As shown in [22], a reduction of network traffic in the order of 30% could be theoretically achieved by compression techniques, while not inflating the false positive ratio. However, this neglects the issue of the excessive demand for memory resources.

These concerns cannot be addressed by merely selecting low capacity BFs. Selecting $C_{BF} << s$ increases the number of BFs maintained by the node, correspondingly leading to increased look-up processing overheads (see Equation 9). Moreover, lower C_{BF} quire a higher number of bits-per-element to support the R_{max}^{Node} constraint. Figure 1 shows the bits-per-element required for various example C_{BF} values, according to Equation 5. At the same time, the increase of the bits-per-element ratio, further increases the processing overheads as it corresponds to a higher number of hash functions to be applied to the IO name included in an incoming name resolution request.

In all, it becomes apparent that for a certain number of IO names, s, the selection of a BF configuration should strike a balance between the resulting memory and processing resource requirements. At first glance, selecting $C_{BF} = s$ offers the best compromise between memory and processing overheads. However, as we show in the following, the need for a single, globally fixed BF configuration, along with the highly skewed distribution of state across the inter-domain topology, render the selection of a BF configuration particularly hard.

4. INTERNET-WIDE BF-BASED NRS

In this section, we empirically explore the range of gains that can be obtained through different BF configurations, for different types of ASes. We engage in a detailed, quantitative study of the distribution of name resolution state in the inter-domain topology. Our target is to investigate the identified interplay between the state distribution and the BF configuration, quantifying its impact on the resulting memory and processing resource requirements.

4.1 Methodology

Our study is based on the AS-level Internet graph inferred by the CAIDA BGP traces[10] [1]. We classify the ASes appearing in the CAIDA trace set into four tiers based on the

[10]The traces are created by first collecting traceroute-like IP-level topology data from several vantage points in the Inter-

size of their customer *cone*, i.e., the total number of their downstream customers [23]. The four tiers are:

1. *Stub* networks, i.e., domains with no more than 4 customer networks, which includes all access networks. As shown in Table 2, these networks constitute the large majority (93.38%) of ASes in the Internet,

2. *Small ISPs*, i.e., small Internet Service Provider ASes that have a cone size between 5 and 50,

3. *Tier-1* ASes, i.e., ASes at the highest level of the hierarchy that do not act as customers for another AS,

4. *Large ISPs*, a category which includes the remaining ASes that have a larger cone than small ISPs but are not tier-1 members.

We parse this graph with a custom, Java-based DONA / CURLING simulator which simulates the registration of IOs across the domains by traversing the appropriate parts of the graph. We first assume that content is uniformly distributed across content providers, which uniformly reside at the leafs of the inter-domain hierarchy. Table 2 provides the average and median percentage of total IO entries held by each AS across all tiers and per tier [18]. We notice a heavily skewed distribution of state across the topology tiers. For instance, more than 93% of the ASes are expected to maintain only up to 0.003% of the total IO entries, in CURLING, while only a small set of tier-1 ASes are burdened with the entire name resolution state in the Internet, in the case of DONA. Moreover, by not exchanging state across peering links, CURLING results in a considerably lower state overhead, especially at the lower tiers. We next utilize the derived results to assess the impact of different BF configurations on the resulting resource requirements.

4.2 Empirical observations

4.2.1 Gains

The heavily skewed distribution of IO entries across the inter-domain topology, results in a correspondingly skewed distribution of resource requirements. Figures 2 and 3 show the memory (M) and processing (LO) resource requirements of DONA-BF and CURLING-BF schemes, for various BF configurations, reaching a maximum $C_{BF} = S$. The figures further illustrate the corresponding requirements of DONA and CURLING.

As expected, memory requirements are in all cases of C_{BF} lower for tier-1 ASes, for both DONA-BF and CURLING-BF, against their non-BF counterparts, since $C_{BF} < s \leq S$. In the extreme case of $C_{BF} = S$, a single 23.96 TB BF would be required to represent all states at all tiers of the topology, for $R^{Node}_{max} = 10^{-4}$. This corresponds to approximately 94,29% decrease in the size of the maintained state at tier-1 ASes, compared to DONA (90,25% in CURLING). Though this reduction confirms the initial motivation for employing BFs, it also questions the practical feasibility of such BF configurations. Supporting efficient lookup operations for name resolution requests requires the queried BF(s) to reside in main memory (instead of storage), leading to limitations by current hardware capabilities. Typical data center servers are configured with only some tens on GB of RAM, constituting a cloud based approach infeasible. Current high end servers, on the other hand, have only just started reaching such capacities[11].

At the same time, selecting such large sized BFs raises concerns about the memory requirements at lower tier ASes which present substantially lower volumes of IO name registrations (i.e., $C_{BF} >> s$). We see that in both cases of DONA-BF and CURLING-BF, large C_{BF} values result in memory resource requirements that exceed those of plain DONA and CURLING, thus providing no incentives for the vast majority of ASes to adopt a BF-based approach. A closer inspection shows that preserving these incentives across the inter-domain topology, in terms of memory requirements, calls for BFs smaller than $C_{BF} = 2^{31}$, $m = 9.86$ GB (we denote this upper limit as C^{max}_{BF}). For instance, the use of 718.88 KB BFs (i.e., $C_{BF} = 10^5$) would result in only an average of approximately 1.95 GB of total memory space for the BF-based representation of state at stub domains, in CURLING-BF, against 359.43 GB, for $C_{BF} = 10^{11}$, and 11.37 GB when not using BFs.

On the other hand, as demonstrated in Figures 2 and 3, using multiple, lower size BFs so as to limit the memory requirements of lower tier ASes, increases the number of lookup operations per name resolution request. For instance, each resolution request reaching a tier-1 AS in DONA-BF, using $C_{BF} = 10^5$ and $m = 718.88$ KB, would require an average of 8.39×10^8 BF lookup operations, against a load in the order of several tens of operations for DONA. This observation becomes especially important when considering that a major fraction of name resolution requests are actually served by tier-1 ASes [18]. On a closer inspection, preserving these incentives across the inter-domain topology, in terms of processing requirements, calls for BF larger than $C_{BF} = 2^{37}$, $m = 482.75$ GB (we denote this lower limit as C^{min}_{BF}).

It follows that a BF configuration lowering both the memory and processing requirements should fall into the [C^{min}_{BF} .. C^{max}_{BF}] range. However, we observe that $C^{min}_{BF} > C^{max}_{BF}$, which means there is no single BF configuration that can yield both lower memory and processing resource requirements for all ASes in the inter-domain topology, compared to a BF-agnostic scheme.

4.2.2 Costs

Since all BF configurations result in the wastage of resources at some areas of the inter-domain topology, we are next interested in identifying the extend of the wastage as well as the areas of the inter-domain topology it affects most. Our target is to get a better understanding on how much the various BF configurations affect the incentives of stakeholders to invest on a BF-based NRS, considering both types of memory and processing resources simultaneously. Towards this end, we first express the wastage (w) brought to an AS by a BF configuration for each type of resource compared to plain DONA or CURLING, as follows:

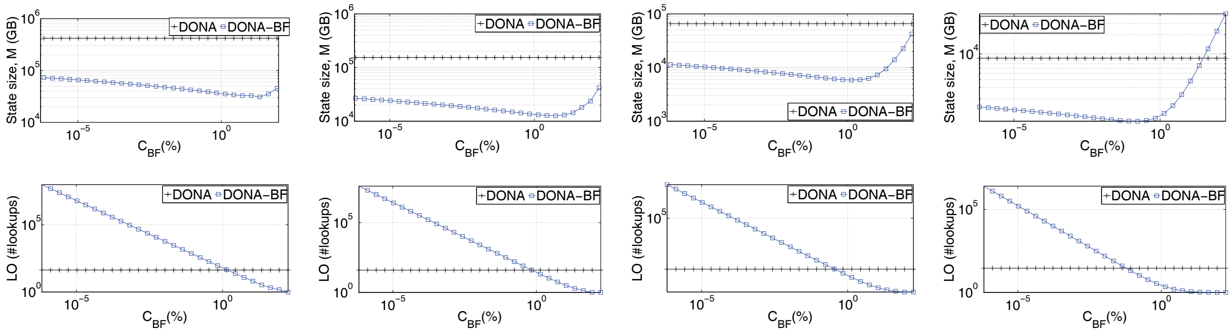
$$w^{y}_{i,x-BF} = max\{0, 1 - \frac{y^{x}_{i}}{y^{x-BF}_{i}}\} \quad (11)$$

net and subsequently identifying the involved ASes from the traced IP addresses.

[11]See for instance: https://www.oracle.com/servers/sparc/m6-32/index.html

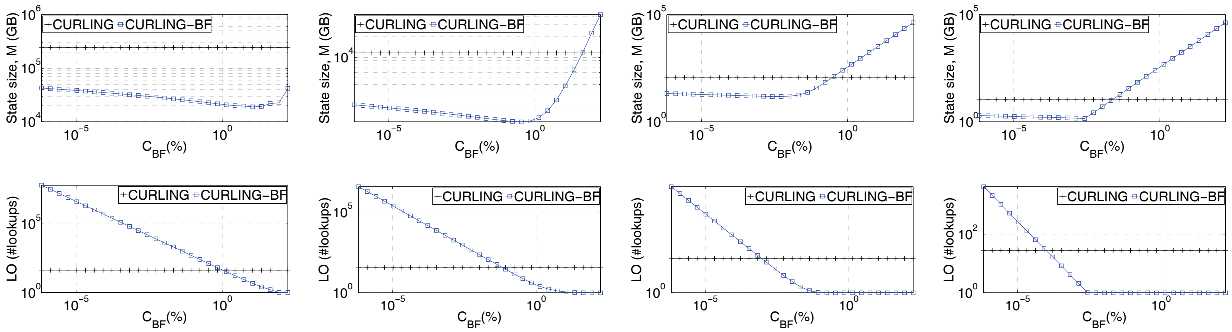
		DONA		CURLING		
Type	(%)	**Average**	**Median**	**Average**	**Median**	**Avg. gain**
All Tiers	100.00%	3.778%	0.003%	0.060%	0.003%	62.97%
Tier-1	0.03%	100.00%	100.00%	59.895%	61.769%	1.67%
Large ISP	1.35%	36.701%	42.687%	2.758%	0.298%	13.31%
Small ISP	5.23%	15.599%	0.097%	0.029%	0.018%	537.90%
Stub	93.38%	2.039%	0.003%	0.003%	0.003%	679.67%

Table 2: State size per AS expressed as a percentage of the total state size throughout the inter-network (%).



(a) Tier-1 ISPs (b) Large ISPs (c) Small ISPs (d) Stub ISPs

Figure 2: Comparison of memory and processing overheads between DONA and DONA-BF for different BF configurations ($R^{Node}_{max} = 10^{-4}$, $S = 10^{13}$). Here C_{BF} is expressed as a percentage of the total number of IOs in the inter-network. Note that both axes are in log-scale.



(a) Tier-1 ISPs (b) Large ISPs (c) Small ISPs (d) Stub ISPs

Figure 3: Comparison of memory and processing overheads between CURLING and CURLING-BF for different BF configurations ($R^{Node}_{max} = 10^{-4}$, $S = 10^{13}$). Here C_{BF} is expressed as a percentage of the total number of IOs in the inter-network. Note that both axes are in log-scale.

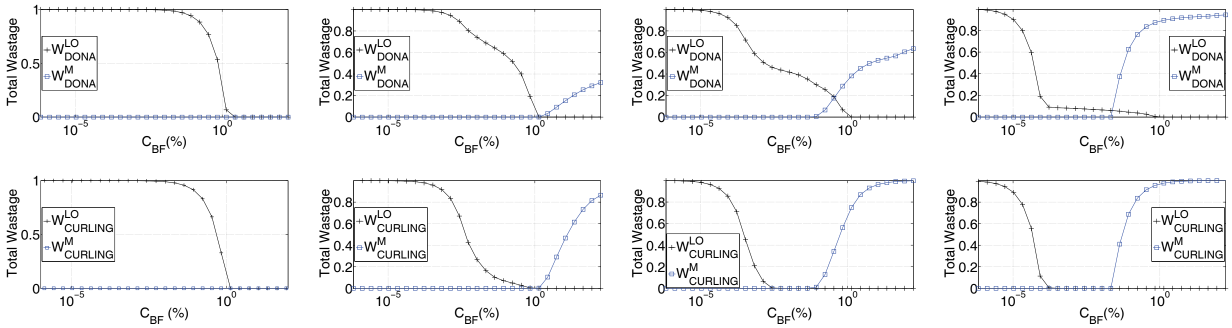
where $y \in \{M, LO\}$,$x \in \{DONA, CURLING\}$, i is the index of an AS, with $i \in [1,..,N]$, and N denotes the total size of the AS population. Equation 12 expresses the normalised total waste of a BF configuration.

$$W^y = \frac{\sum_{i=1}^{N} w^y_{i,x-BF}}{N} \tag{12}$$

Figure 4 depicts the total wastage W at the various tiers of the inter-domain topology in the case of DONA-BF and CURLING-BF. We first see that, for both DONA-BF and CURLING-BF, resource wastage at Stub networks is reduced across both types of resources, for C_{BF} values in the range of 10^{-4} to 10^{-2}% of the total IO catalogue size (S), e.g., from $C_{BF} = 2^{23}$, $m = 50.63$ MB to $C_{BF} = 2^{32}$, $m = 19$ GB for $S = 10^{13}$. This constitutes the vast majority of the ASes in the Internet, thus motivating the selection of a BF configuration in this range. Nevertheless, at this range, we notice a substantial wastage of processing resources for tier-1 ASes, with Large and Small ISPs also being substantially affected in the case of DONA-BF. In all, it becomes evident that even if a subset of the ASes in the Internet is willing to tolerate the wastage of some type of resource, over the other (e.g., Stub networks tolerate some increased processing resource requirements in order to reduce their memory costs in the aforementioned range), still, this would have a substantial impact in other areas of the network, particularly dis-incentivising the corresponding ASes (e.g., tier-1 ASes). Evidently, this observation renders the selection of a globally fixed BF configuration infeasible.

5. SIMULATION EVALUATIONS

Though enabling the identification and assessment of the tradeoffs emerging in the design and configuration of the considered BF-based NRS approaches, the methodology employed in the previous section presents some limitations.



(a) Tier-1 ISPs (b) Large ISPs (c) Small ISPs (d) Stub ISPs

Figure 4: Total wastage observed across the inter-domain topology tiers, for DONA-BF and CURLING-BF.

First, our analysis is based on the simplifying assumption of a perfect allocation of IO name registrations into BF containers, while the actual number of BFs depends on the merging process throughout the hierarchy (see Section 3). Second, the assessment of the processing overheads calls for a closer look at the actual number of name resolution requests reaching each node. Third, our analysis does not provide insights on the overall impact of false positives on the entire inter-domain topology, as it focuses on a per node upper limit, i.e., R^{Node}_{max}.

In this section, we address these limitations by resorting to packet level simulations, which enable the investigation of the dynamics of the name resolution system and the actual impact of the complex inter-domain topology structure on the identified performance metrics.

5.1 Simulation environment

As the size of inter-domain topology graph presents significant scalability challenges for our simulation environment (i.e., more than 45K ASes and approximately 200,000 annotated links [1]), we employ scaled-down inter-domain topologies generated by the algorithm proposed by Dimitropoulos et *al.* [7], which present a manageable size for our evaluation purposes while maintaining the same properties of the original CAIDA graph. Specifically, we employ a topology of 400 ASes inter-connected with multi-homing and peering links. On top of this topology, we deploy one RH node per AS. We neglect intra-domain communication overheads and focus our study on the effects of the inter-domain topology structure on the performance of the considered NRSes. In these scaled-down topologies, we classify the ASes according to their minimum hop-count distance to the top level of the hierarchy. Finally, we use a synthetic workload which considers a detailed mixture of various traffic types (e.g., Web, Video, P2P). For this purpose, we employed the GlobeTraff traffic generator tool [16], to generate workload instances with an average IO catalogue size of 79,821 items and 190,065 resolution requests. Each simulated scenario consists of two phases. In the first phase, the entire catalogue of IOs is registered to the NRS. Then, the entire set of resolution requests is injected into the network from randomly selected leaf ASes (i.e., ASes with an empty cone). We consider various values of C_{BF} ranging from $C_{BF} = 10$ to $C_{BF} = 10^5$, and set $R^{Node}_{max} = 10^{-4}$.

5.2 Results

Based on the described simulation environment, we first validate the methodology and findings presented in the previous sections. To this end, we assess the distribution of the states across the inter-domain topology, along with processing overheads as defined in Section 3.3.3.

State size. Figure 5 shows the average state size for all considered schemes and various BF configurations. We see that the state size varies in a similar pattern to our analytical model, with large C_{BF} values eventually leading to state sizes that exceed those observed for plain DONA and CURLING.

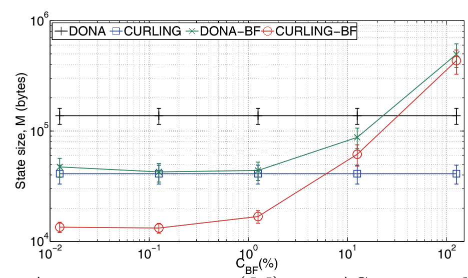
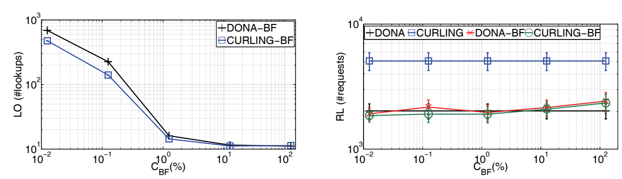


Figure 5: Average state size (M) per AS across the inter-domain topology.

Processing overheads. Figure 6(a) shows the average number of lookup operations (LO) per received name resolution request, across all ASes. As previously described, this metric expresses the processing overheads associated with each request. As we see, these overheads are directly related to the BF configuration, presenting a trend corroborating our analytical model.



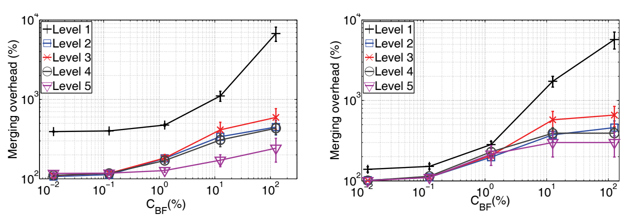
(a) LO (b) RL

Figure 6: Processing overheads for various C_{BF} values. In 6(a): average LO per AS across the inter-domain topology. In 6(b): average RL per AS across the inter-domain topology.

In Figure 6(b), we further show the average total number of name resolution requests received at each AS in the topology (i.e., RL, see Section 3). We notice that CURLING presents the highest load among the considered schemes. As shown in [18], this is because no state is exchanged over peering links in CURLING, resulting in name resolution requests being propagated further up in the hierarchy. Both BF-based schemes present a performance similar to DONA, with slightly higher overheads in the case of the largest C_{BF} values. At such values, BFs are sparser and thus present a lower false positive ratio, thus allowing requests to further propagate in the inter-domain topology (see also Figure 8). In all, Figure 6(b) suggests that the BF configuration has a minor impact on the total number of resolution requests received by each AS.

BF merging efficiency. As mentioned above, our assumption on perfect assignment of IO registrations to BFs only supports a baseline understanding of the identified trade-offs. In order to assess the efficiency of a realistic merging process we resort to simulations. In our model we employ the *Best First Decreasing* bin-packing algorithm. We consider an offline algorithm since we assume a periodic state update mechanism (see Section 2).

We measure the efficiency of the bin-packing algorithm with the help of the *merging overhead* metric, which corresponds to the ratio of the actual number of BFs over the theoretically optimal F. Figures 7(a) and 7(b) present this overhead for each inter-domain topology level, for DONA-BF and CURLING-BF respectively. In both cases, large C_{BF} values result in considerable overheads at all layers with the highest impact on Level 1 ASes, i.e., tier-1. A close inspection of the inter-domain topology structure reveals that this is the direct consequence of a large number of ASes being directly connected to tier-1 ASes. Specifically, we observe that approximately 48% of the leaf nodes in the entire topology are directly connected to some tier-1 AS, corresponding to ASes that aim to minimize transit overheads [20]. This has the effect of multiple BFs arriving directly to tier-1 ASes without having been previously merged with each other at some intermediate AS. It is noted that tier-1 ASes do not merge incoming BFs from multiple customer ASes as this would not allow them to preserve forwarding information.



(a) DONA-BF (b) CURLING-BF

Figure 7: Average merging overhead across the inter-domain topology levels for various C_{BF} values.

Overall False Positive Ratio. Figure 8 shows the overall false positive ratio observed across the hierarchy, i.e., the ratio of the total number of name resolution requests that suffered a false positive response, over the entire population of name resolution requests in our workload (denoted as DONA-/CURLING-BF (All)). Interestingly, we see a considerably high false positive ratio in the order of 90%. To get a better understanding of this result, we first note that, following widely studied popularity models, our workload included multiple requests for the most popular IOs [16]. This means that if a request leads to a false positive at a certain node, then all subsequent requests for the same IO at the

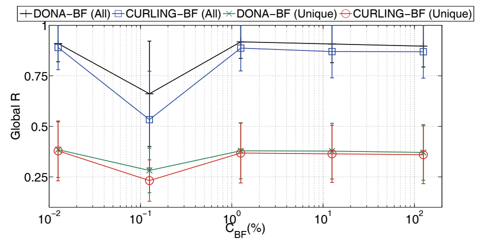
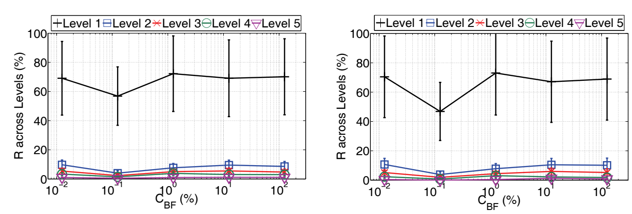


Figure 8: Average false positive ratio (R) across the inter-domain topology.

same node will deterministically lead to false positives.[12] To further assess the impact of IO popularity on the perceived false positive ratio, Figure 8 further depicts the observed values for an alternative workload with a unique request per IO in the network (denoted as DONA-/CURLING-BF (Unique)). This removes the effect of repeated false positive events for the same resolution requests (i.e., the overall rate of false positive events) and reflects the false positive probability. We notice a substantial reduction to values in the order of 40%, which are still considered impractical.

To further shed some light on this finding, we inspect the distribution of the observed false positive events across the levels of the inter-domain topology. Figures 9(a) and 9(b) show that in both cases of DONA-BF and CURLING-BF, the vast majority of false positives takes place at tier-1 ASes. This is a direct outcome of the observed concentration of large numbers of BFs at tier-1 ASes from directly attached leaf ASes. Though the R_{max}^{Node} constraint is enforced through the configuration of BFs, the structure of the inter-domain topology results in a volume of BFs that increases the overall probability of suffering a false positive (see Appendix). Moreover, it must be noted that the overall false probability ratio is further increased by the multi-hop resolution process i.e., the false positive ratio at each node has a cumulative effect.



(a) DONA-BF (b) CURLING-BF

Figure 9: Distribution of average False Positive ratio across the levels of the inter-domain topology.

6. CONCLUSIONS AND FUTURE WORK

Promising a compact representation of name resolution state, BFs have been considered as a potential solution towards the scalability concerns stemming from the enormous size of the information namespace. In this paper, we have attempted to shed some light on the properties of BF-based inter-domain name resolution schemes, focusing on the resulting resource requirements and the interplay between the

[12]This assumes that between consecutive name resolution requests, the queried BFs along the name resolution path remain unchanged. We leave this issue for future investigation.

BF configuration and the distribution of state across the inter-domain Internet topology. Our findings show that the skewed distribution of name resolution state renders the selection of a BF configuration particularly problematic. According to our findings: (i) it is not possible to select a BF configuration such that both the memory and processing resource requirements are reduced for all ASes, compared to BF-agnostic NRSes, (ii) reducing memory resource requirements for the majority of ASes in the Internet, inevitably inflates processing resource requirements at tier-1 ASes, (iii) the multiplicity of customer-provider routing relationships between tier-1 and stub networks inflates the number of BFs maintained by the former, dramatically increasing false positives, and thus deteriorating the reliability of the NRS.

These results directly question the feasibility of BF-based inter-domain NRSes. The resulting resource requirements lead to a conflict of interests in the inter-domain topology with ASes at different tiers presenting contradictory requirements. Even worse, the structure of the inter-domain topology results in inflated false positive ratios that question the reliability of the name resolution process.

In view of these conclusions, the investigation of alternative probabilistic data structures appears as the next step. To this end, a series of BF enhancements and alternative structures has been proposed, offering some appealing properties [24]. Scalable BFs [2] and Dynamic BFs [13] target at dynamically adapting to varying and/or potentially large set sizes. Counting BFs (CBFs) [10] and d-left CBFs [4] further enable the deletion of registered items, thus avoiding the periodic update of the entire name resolution state. Finally, the Cuckoo [9] filter was recently proposed as a BF alternative presenting lower memory and processing requirements, while reducing false positive rates.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7-ICT-2011-8 under grant agreement no. 318708 (C-DAX) and the CHIST-ERA / EPSRC UK project CONCERT (grant no. EP/L018535/1). The authors alone are responsible for the content of this paper.

7. REFERENCES

[1] Caida dataset. http://www.caida.org/research/topology/#Datasets, 2008. [Online; accessed 19-July-2008].

[2] P. S. Almeida et al. Scalable bloom filters. *Information Processing Letters*, 101(6):255 – 261, 2007.

[3] B. H. Bloom. Space/Time Trade-offs in Hash Coding with Allowable Errors. *Commun. ACM*, 13(7):422–426, July 1970.

[4] F. Bonomi et al. An improved construction for counting bloom filters. In Y. Azar and T. Erlebach, editors, *Algorithms âĂŞ ESA 2006*, volume 4168 of *Lecture Notes in Computer Science*, pages 684–695. Springer Berlin Heidelberg, 2006.

[5] W. K. Chai et al. Curling: Content-ubiquitous resolution and delivery infrastructure for next-generation services. *IEEE Communications Magazine*, 49(3):112 –120, 2011.

[6] C. Dannewitz et al. Hierarchical DHT-based name resolution for information-centric networks. *Computer Communications*, 36(7):736–749, Apr. 2013.

[7] X. Dimitropoulos et al. Graph annotations in modeling complex network topologies. *ACM Transactions on Modeling and Computer Simulation*, 19:17:1–17:29, November 2009.

[8] D. Evans. The Internet of Everything: How More Relevant and Valuable Connections Will Change the World. White Paper, February 2012.

[9] B. Fan et al. Cuckoo filter: Practically better than bloom. In *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, CoNEXT '14, pages 75–88, New York, NY, USA, 2014. ACM.

[10] L. Fan et al. Summary Cache: a scalable wide-area web cache sharing protocol. *IEEE/ACM Transactions on Networking*, 8(3):281–293, 2000.

[11] N. Fotiou et al. H-Pastry: An inter-domain topology aware overlay for the support of name-resolution services in the future Internet. *Computer Communications*, 62:13 – 22, 2015.

[12] Google. *We knew the web was big, July 2008.*

[13] D. Guo et al. The dynamic bloom filters. *Knowledge and Data Engineering, IEEE Transactions on*, 22(1):120–133, Jan 2010.

[14] J. Hong et al. Bloom Filter-based Flat Name Resolution System for ICN. Internet-Draft draft-hong-icnrg-bloomfilterbased-name-resolution-03.txt, IETF Secretariat, Mar. 2015.

[15] R. Jarno et al. On name-based inter-domain routing. *Computer Networks, Elsevier*, 55:975–986, March 2011.

[16] K. Katsaros et al. GlobeTraff: A Traffic Workload Generator for the Performance Evaluation of Future Internet Architectures. In *New Technologies, Mobility and Security (NTMS), 2012 5th International Conference on*, pages 1–5, May 2012.

[17] K. V. Katsaros et al. On inter-domain name resolution for information-centric networks. In *Proc. of the IFIP TC 6 Conference on Networking*, pages 13–26, 2012.

[18] K. V. Katsaros et al. On the Inter-domain Scalability of Route-by-Name Information-Centric Network Architectures. In *Proc. of the IFIP TC 6 Conference on Networking*, pages –, 2015.

[19] T. Koponen et al. A data-oriented (and beyond) network architecture. In *Proc. of the ACM SIGCOMM*, pages 181–192, 2007.

[20] C. Labovitz et al. Internet inter-domain traffic. In *Proc. of the 2010 ACM SIGCOMM*, pages 75–86, New York, NY, USA, 2010. ACM.

[21] H. Liu et al. A multi-level DHT routing framework with aggregation. In *Proc. of the 2012 ACM SIGCOMM Workshop on Information-centric networking (ICN'12)*, pages 43–48. ACM, 2012.

[22] M. Mitzenmacher. Compressed Bloom Filters. *IEEE/ACM Trans. Netw.*, 10(5):604–612, Oct. 2002.

[23] R. Oliveira et al. The (in)completeness of the observed internet AS-level structure. *IEEE/ACM Transactions on Networking*, 18:109–122, February 2010.

[24] S. Tarkoma et al. Theory and practice of bloom filters for distributed systems. *Communications Surveys Tutorials, IEEE*, 14(1):131–155, First 2012.

[25] G. Xylomenos et al. A Survey of Information-Centric Networking Research. *IEEE Communications Surveys and Tutorials*, 16(2):1024–1049, 2014.

APPENDIX

As discussed in Section 5, the direct connection of multiple leaf ASes to Tier-1 ASes results in the latter maintaining a large number of BFs, considerably sparse in the case of large C_{BF} values. In turn the false positive rate at Tier-1 ASes is inflated. This is because for a certain BF configuration and a certain number of IOs s, the use of multiple sparse BFs increases the effective false positive ratio compared to fewer but denser BFs. Considering a positive integer $\beta \in \mathbb{Z}$:

$$\begin{aligned} s < \beta s &\Leftrightarrow \\ R(s) < R(\beta s) &\Leftrightarrow \\ (1 - R(s))^{\beta} < (1 - R(\beta s))^{\beta} &\Leftrightarrow \\ 1 - (1 - R(s))^{\beta} > 1 - (1 - R(\beta s))^{\beta} > 1 - (1 - R(\beta s)) &\Leftrightarrow \\ R^{Node}(s, \beta) > R(\beta s) & \end{aligned}$$

where $R(s)$ is the false positive ratio of a BF with s elements inserted and $R^{Node}(s, \beta)$ is the false positive probability at a node maintaining β BFs of s elements each.