A Privacy-Preserving Collaborative Caching Approach in Information-Centric Networking

Andrew Jones(✉) and Robert Simon

George Mason University, Fairfax, VA 22030, USA
{ajones93,simon}@gmu.edu

Abstract. It has been established that in-network caching in an Information-Centric Network (ICN) environment significantly reduces required bandwidth and content retrieval delay, and reduces load on content producers. However, malicious actors masquerading as legitimate consumers can probe cache contents and use the resultant data to map content objects to, and thereby violate the privacy of, the consumer(s) who requested them. Existing mitigation approaches suffer a direct trade-off between privacy and utility; the two are diametrically opposed, and prioritizing either rapidly degrades its counterpart. This paper presents a *collaborative caching* approach with provable privacy and utility guarantees that instead monotonically increase as a function of one another, growing in tandem. Our proposed scheme preserves all true cache hits to utilize in-network caching as efficiently as possible. We have evaluated our method against a number of other in ICN caching policies for a variety of workloads and topologies. Our results show that our technique delivers high cache hit ratios and minimizes interest satisfaction delay while offering provable privacy guarantees.

Keywords: Secure information centric networking · Provable privacy · Distributed system security

1 Introduction

Information-Centric Networking (ICN) encompasses a paradigm shift from the point-to-point, address-based IP protocol which comprises the "thin waist" of today's internet. ICN eschews this existing model in favor of an architecture in which content is treated as a first-class citizen and is named, addressable, and routable [6]. At a high level, entities within an ICN are content producers, content consumers and routers. ICN development is motivated by modern internet usage patterns resembling those of a content distribution network (CDN). IP was designed to address the needs of a network of hosts intercommunicating via relatively equally-weighted full duplex conversations. However, many of the hosts in today's internet operate almost exclusively as consumers, requesting

S. Devismes and N. Mittal (Eds.): SSS 2020, LNCS 12514, pp. 133–150, 2020.
https://doi.org/10.1007/978-3-030-64348-5_11

content from those who produce it. ICN is the product of an attempt to design an internet architecture better suited to this model of communication.

An important feature of proposed ICN architectures is the utilization of in-network content caching at routers. However, if implemented in a naive fashion, ICN content caching is susceptible to attacks against consumer privacy. In this context consumer privacy is informally defined by asserting that a legitimate consumer (Alice) wishes to hide the fact that she has requested a content object $\mathcal{O}$. Suppose a malicious user (Darth) connects to same first-hop router $\mathcal{R}$ to which Alice is connected, and wants to determine if, indeed, Alice has requested $\mathcal{O}$. As described in greater detail in Sect. 2.1, Darth issues a request for $\mathcal{O}$. Darth also has determined the expected time $T_{\mathcal{R}}$ to satisfy content requests from $\mathcal{R}$. If the time to receive $\mathcal{O}$ is approximately equal to $T_{\mathcal{R}}$ then Darth can conclude that Alice has previously requested $\mathcal{O}$. Note that this attack still works if users besides Alice are connected to $\mathcal{R}$.

Defenses against the attack just described include TOR-like mechanisms or introducing artificial delays to request response. Both of these approaches introduce performance penalties. The contribution of our paper is to introduce a *collaborative caching* policy designed to defeat consumer privacy attacks without introducing significant performance penalties. We focus on domain-clustering ICN deployments and show how to serve a content request from an in-network cache in such a way as to hide from Darth information about Alice's content requests. We also show that our scheme produces a provable privacy bound, in the sense of providing (ϵ, δ)-probabilistic indistinguishability, a standard measurement used to quantify the utility of privacy protocols [8,16].

2 Background and Related Work

Over the last decade or so there have been several proposed Information Centric Networking architectures, such as the European PURSUIT project [5]. Our work is motivated by research in Content Centric Networking proposals and work in the ongoing Named Data Networking (NDN) project [10,19].

2.1 NDN Overview

Content retrieval in NDN[1] does not necessitate a persistent end-to-end connection between the entity which produced it and that which is requesting it. Rather, network endpoints fall into one or both of the following categories: **consumers**, which issue *interests* for the data they wish to retrieve, and **producers**, which dispatch *content* packets to *satisfy* received interests. Notably, a host in the network can be both a producer and consumer. *Pure* consumers or producers are those which perform solely the functions of consumers or producers, respectively. A pure consumer has no addressable namespace and no private/public key pair for signing and authenticating its (nonexistent) content. Content packets in NDN

[1] After this section we revert to the abbreviation ICN.

are only forwarded to consumers which specifically request them via interests. A noteworthy security implication of these policies is that pure consumers are not addressable entities in ICN. Two data structures are present in each router in an NDN network: A **Pending Interest Table** (PIT), which records each of the interests which have arrived at the router and the corresponding interfaces on which they were received, and a **Forwarding Interest Base** (FIB), which contains a mapping of content name prefixes to outgoing interfaces.

A router is not required to but may additionally possess a *content store* (CS). A router can opportunistically cache content in its CS, upon receiving that content in response to a previously forwarded interest. The router can then serve that content from its CS in response to future interests received for the same content. This has the benefit of potentially greatly reducing data retrieval delays, as an interest and its corresponding content may not need to traverse the entire path between a consumer and producer. Any content received by a router which does not match an entry in the router's PIT is discarded. The focus of our work is to ensure consumer privacy in the face of timed probing attacks against content stores. In NDN the content that satisfies an interest is always forwarded along the reverse path of the interest which requested it. The determination of this reverse path in the absence of a source address is accomplished by per-interest state recorded at each router hop in the form of an entry in the PIT. Upon receipt of an interest for the same content as another interest already in its PIT, a router will simply add the interface on which the new interest was received to the existing entry in its PIT and discard the remaining information in the new interest without forwarding it. Corresponding content is returned along all necessary interfaces whilst avoiding duplication. Producers and/or other routers are not inundated with multiple interests forwarded by a given router requesting the same content. The satisfaction of a single interest by a producer may serve the content in question to many consumers whose interests were collapsed into that received by the producer.

2.2 Related Work

There has recently been significant interest in ICN cache privacy issues. ANDaNA [4] and AC3N [25] are applications of the onion routing and ephemeral circuits of TOR to ICN. Though effective, these approaches increase latency, decrease available bandwidth compared to vanilla NDN, and- due to ephemeral encryption- prohibit any useful in-network caching.

A proposed mitigation to the cache privacy attack which incorporates a randomized content satisfaction delay to mask cache hits is presented in [2]. A router R can introduce an artificial delay before responding to an interest. In doing so, R prevents an adversary $\mathcal{A}$ from determining whether or not a given piece of content C is in its content store (CS), denoted CS_R. This work also establishes privacy bounds. As with all approaches that introduce artificial delays, performance can become an issue. Somewhat similar to [2] is the work presented in [18] and [17] which uses privacy preserving delays at edge routers. The work described in [1] uses the concept of "Betweenness Centrality-based" caching that caches content at nodes with a higher betweenness centrality value to put consumers in larger anonymity sets. Unlike this work, we focus on providing consumers with a uniform anonymity level, and we provide a computable privacy bound. Additional related efforts include a namespace-based privacy [14] approach, and a Long Short Term Memory detection approach to detect a timing attack [27]. The work described in [26] details an edge-based access control mechanism for ICNs. Our work differs from the above papers because we focus on protecting individual consumers without sacrificing performance, and because we offer a provable privacy bound. We also note that our methods are not vulnerable to attacks that exploit hop limit and scope fields in the NDN packet header [2].

We note that there is recent interest in using domain clustering methods, in conjunction with hash routing, to support large ICNs [23]. Our approach relies on the use of clustering.

3 Collaborative Caching Algorithm

As illustrated in Fig. 1a, our proposed caching scheme divides the network into clusters of routers, each of which will operate as a distributed aggregate in-network cache. This abstraction is transparent to producers, consumers, and other clusters. The cluster to which a router will belong is determined by the partitioning around medoids algorithm [12]. Upon the arrival of a content packet at a router on the edge of a cluster, a router is chosen uniformly at random from the members of the cluster (including the specific router which actually received the content packet) as the designated, or "authoritative", router at which to cache the content. The content is then multicast to both the designated router cache for later use and to the next-hop router on the path back to the consumer which originally issued the interest for the content.

When a router on the edge of a cluster encounters an interest, that interest is forwarded to the authoritative router cache pertaining to the content requested by the interest, if one has already been determined. If the requested content is in the content store of a router in the cluster, it is returned to the consumer which issued the interest. If not, a single interest for the entire cluster is propagated upstream toward the appropriate producer by the cluster router closest to that producer. This process is illustrated in Fig. 1b and detailed in Algorithm 1.

We now describe our system and adversary model. Let Σ^* and Γ denote the universes of all content names (composed of some finite alphabet Σ) and content objects, respectively, using notation in common with [2]. Let G represent a cluster of collaborating routers according to our proposed caching model, and U represent the set of all consumers downstream from G. $S : (\Gamma, U) \rightarrow \mathbb{N}$ represents, for a given content item $C \in \Gamma$ in the cache of any router in G, the number of times C has been forwarded by G to $u \in U$. Note that we use the definition of $\mathbb{N}$ from ISO 8000-2 [9], where $0 \in \mathbb{N}$. $S(C, u) = 0$ for all content not in any router cache in G, and for all content for which u has not issued an interest.

We allow consumers to determine whether specific content has been forwarded by G via probing attacks. As in [2], this is modeled by a function

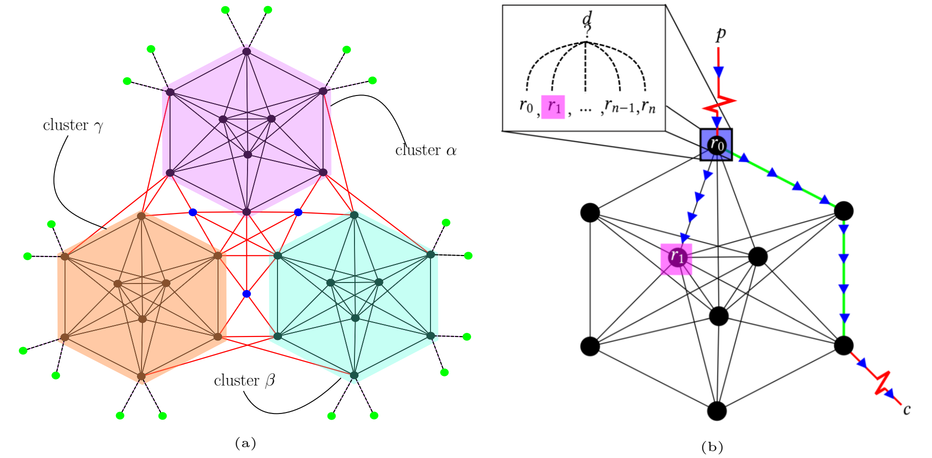


Fig. 1. (a) A network divided into router clusters α, β, and γ, the routers belonging to which lie within the pink, cyan, and orange shaded regions, respectively. Producers are denoted by blue nodes, consumers by green nodes, and routers by black nodes. Black solid edges represent intra-cluster connections, whereas red edges are inter-cluster or cluster-to-producer connections and consumer-to-router connections are indicated with dashed black edges. (b) Caching process within a cluster. A content packet d, requested by consumer c, has just arrived at router r_0 from producer p. r_1 is selected uniformly at random from all cluster routers to cache d. Blue arrowheads indicate the multicast flow of d to router r_1 and along the (green) shortest path to c. (Color figure online)

$Q_S : \Sigma^* \rightarrow \{0, 1\}$. We let $N_C \in \Sigma^*$ denote the name associated with content $C \in \Gamma$. In network state S,

$$Q_S(N_C, G) = \begin{cases} 1, & \text{if cached content } C \text{ in } G \text{ matches the input name } N_C \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Each invocation of $Q_S(N_C, G)$ by a given consumer u causes S to transition to S' such that:

1. $S'(C, u) = S(C, u) + 1$
2. $\forall C' \in \Gamma \setminus \{C\}, S'(C', u) = S(C', u)$
3. $\forall C'' \in \Gamma, u' \in U \setminus \{u\}, S'(C'', u') = S(C'', u')$

The attack we are concerned with operates as follows [7]: A malicious consumer, $\mathcal{A}$, is connected to an edge router $\mathcal{R}$, the only other consumer connected to which is u. $\mathcal{A}$ determines the round-trip time to $\mathcal{R}$ by issuing two identical interests with the same random content name and observing the content return delay. $\mathcal{A}$ then issues an interest for some content C and measures that content retrieval delay. If that content (C) is returned with a delay approximately equal to the round trip time (RTT) from $\mathcal{A}$ to $\mathcal{R}$, $\mathcal{A}$ concludes that u recently requested C, as the interest must have been satisfied at the first hop router.

Algorithm 1: Collaborative-Caching

```
Input: Interest $I$ from consumer $N$, requesting content $C$ produced by $P$, Collaborating
       router cluster $G$
Output: $C$, $x = \begin{cases} 1, & \text{if collaborative cache hit} \\ 0, & \text{otherwise} \end{cases}$
$CS_{loc}$ := local content store of router $R$ receiving interest;
if $C \notin CS_{loc}$ then
    if $C$ in $CS_g$ for some $g \in G$ then
        Route $I$ to authoritative router for $C$;
        Return $(C, 1)$ when $C$ returned from authoritative router;
    else
        Decrement HopLimit;
        if HopLimit = 0 then
            Return $(NULL, 0)$;
        end
        Forward $I$ to router $R_E$ on edge of $G$ and onward toward $P$;
        while $C$ has not arrived at $R_E$ from $P$ do
            Wait;
        end
        Determine authoritative router $g_C \in G$ with Pr $= \frac{1}{|G|}$ (uniformly random);
        $R_E$: (Mulicast) send $C$ to $g_C$ for caching and return $(C, 0)$ on shortest path to $N$;
    end
else
    Return $(C, 1)$;
end
```

4 Provable Privacy

4.1 Quantifying "Privacy"

We derive our understanding of cache privacy from the concept of (ϵ, δ)-probabilistic indistinguishability, which we define using a definition motivated by that provided in [2].

Definition 1 ***(ϵ, δ)-probabilistic indistinguishability.*** *Two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ are (ϵ, δ)-probabilistically indistinguishable if we can divide the output space $\Omega = Range(\mathcal{D}_1) \cup Range(\mathcal{D}_2)$ into Ω_1 and Ω_2 such that, letting $\mathcal{Q}_1$ and $\mathcal{Q}_2$ be random variables with probability distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively,*

1. *for all $O \in \Omega_1$, $e^{-\epsilon} \leq \frac{\Pr[\mathcal{Q}_1 = O]}{\Pr[\mathcal{Q}_2 = O]} \leq e^{\epsilon}$*
2. $\Pr[\mathcal{Q}_1 \in \Omega_2] + \Pr[\mathcal{Q}_2 \in \Omega_2] \leq \delta$

The similarity of distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ is directly proportional to the magnitude of both ϵ and δ. Minimizing the upper bound on both ϵ and δ is therefore desirable when seeking to prove that two distributions are indistinguishable. Suppose we observe a network in two measurable states, represented by $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively. Intuitively, this definition merely implies that, when ϵ and δ are both small, those states are quite similar. This similarity makes it difficult to distinguish between the distributions. If the two distributions were to respectively represent states of the network in which, on the one hand, consumer u had not requested content C, and, on the other, it had, then the difficulty of distinguishing between the two distributions would be directly related to the difficulty of defeating cache privacy. This is the difficulty of mounting a successful attack which can identify the source of an interest based on cached content.

Our definition of (k, ϵ, δ)-privacy is a modified version of that presented in [2], adapted to suit a collaborative caching model.

Definition 2 ***(k, ϵ, δ)-privacy.*** *For all names $n \in \Sigma^*$, subset of content $M \subset \Gamma$, and pairs of states S_0, S_1 such that $S_0(\gamma, u') = S_1(\gamma, u')$ for all $\gamma \in \Gamma \setminus M$ and for all $u' \in U$, and $S_0(C, u) = 0$ and $0 < S_1(C, u) \leq k$ for all $C \in M$ (i.e., S_0 and S_1 differ only on content objects in M) and for consumer u downstream from router cluster G; $Q_{S_0}(n, G)$ and $Q_{S_1}(n, G)$ are (ϵ, δ)-probabilistically indistinguishable.*

Notably, the above definition does not prohibit S_0 and S_1 from differing in terms of the number or distribution of requests made for content C by routers other than u. We allow, but do not require, $S_0(C, u') \neq S_1(C, u'), \forall u' \in U \setminus \{u\}$, and $S_0(C, u')$ and $S_1(C, u')$ could each be zero or positive for any given router $u' \in U \setminus \{u\}$ (as long as $S_1(C, u') \geq S_0(C, u')$).

4.2 Provable Privacy Guarantee

Our approach, like those which employ artificial delay, ultimately serves to prohibit an adversary from learning if a given consumer has issued an interest for a particular piece of content. However, the two methodologies are divergent with respect to the manner in which this is accomplished. *Random-Caching* [2] seeks to conceal the existence of any particular piece of content in a router's cache, assuming that cognizance of the content's presence there would allow an adversary $\mathcal{A}$ to correctly infer that a specific consumer u requested that content. *Collaborative-Caching* (Algorithm 1) decouples the existence of a content item in a router's cache from the implication that a consumer directly downstream from that router issued an interest for that content. We allow an adversary to successfully determine that a consumer downstream from a collaborating router cluster has issued an interest for some specific content- and even the exact router in the cluster at which that content is cached- without revealing the precise identity of the consumer from which the interest originated. We achieve this by maintaining an anonymity set of a specified size for every router downstream from a collaborating router cluster; the size of a router's anonymity set is no longer determined by the number of other consumers which share its first-hop router.

Let $m_{(v,i)}$ denote the number of interests issued by consumer $v \in U$ for all content in state S_i. Let $Q_0(C, r_0)$ and $Q_1(C, r_1)$ denote the output x of Algorithm 1 in states S_0 and S_1, respectively, with C where r_i denotes the set of expected values of the number of interests for C issued by consumers- other than u- downstream from the collaborating router cluster G in each state S_i. That is, $r_i = \{\mathbb{E}(S_i(C, v)), \forall v \in U \setminus \{u\}\}$ where U denotes the set of all consumers downstream from G. Note that we use zero-based array indexing when referring to elements of r in the subsequent formulae.

Theorem 1 *If all cached content is statistically independent, and consumers issue interests for specific content with uniformly random probability, Collaborative-Caching is*

$$\left(\sum_{v=0}^{|U|-1} m_{(v,0)}, \ln|r_0|, \left(1-\frac{1}{|\Gamma|}\right)^{\sum_{v=0}^{|r_0|-1} m_{(v,0)}}\right)-\textit{private.}$$

Proof. Per Definition 2, $S_0(C,u)=0$ and $S_1(C,u)=n$, where $1 \le n \le k$. $Q_0^t(C,r_0)$ and $Q_1^t(C,r_1)$ denote the sequence of outputs produced by Algorithm 1 when executed t consecutive times with C, in states S_0 and S_1 respectively. As noted in [2], the presumed statistical independence of content simplifies this analysis by allowing us to focus on the difference between S_0 and S_1 only as it relates to C- whether or not other content has been requested and by whom is irrelevant. The following probabilistic analyses therefore assume content object independence and leverage the idea that separate requests for content are statistically independent events. Let $\mathcal{Q}_0^t$ and $\mathcal{Q}_1^t$ denote two random variables describing $Q_0^t(C,r_0)$ and $Q_1^t(C,r_1)$ respectively when consumers request content uniformly at random. Each entry in the zero-indexed set r_i will therefore be: $r_i[v]=m_{(v,i)}\cdot\frac{1}{|\Gamma|}$ where $m_{(v,i)}$ denotes the total number of requests (for all content) which have been made by downstream consumer v in state S_i.

We show that, assuming consumers are equally likely to request or not request C, $\mathcal{Q}_0^t$ and $\mathcal{Q}_1^t$ are (and consequently *Collaborative-Caching* is)

$$\left(\ln|r_0|, \left(1-\frac{1}{|\Gamma|}\right)^{\sum_{v=0}^{|r_0|-1} m_{(v,0)}}\right)-\textit{probabilistically indistinguishable}$$

for any C (a corollary of our assumption that content is statistically independent). Note that in our adversarial model, $\mathcal{A}$ has no knowledge of the likelihood that any particular consumer would be interested in a given piece of content. As such, $\mathcal{A}$ possesses no information from which it can extrapolate that the probability distribution of a given consumer's requests is anything but uniform.

The output x (defined in Algorithm 1) of $\mathcal{Q}_1^t$ will be $\{1\}^t$ because, in state S_1, u has already issued at least one interest for C and it is therefore cached at some router in the router cluster. The output of $\mathcal{Q}_0^t$ will be:

$$\mathcal{Q}_0^t=\begin{cases}\{1\}^t, & \text{if } \exists v \in U\setminus\{u\} \text{ s.t. } S_0(C,v)\ge 1\\ 0||\{1\}^{t-1}, & \text{otherwise}\end{cases}$$

That is to say, $\mathcal{Q}_0^t$ will be either $\{1\}^t$ (a sequence of t ones), if a consumer other than u has already issued at least one interest for C in S_0, or $0||\{1\}^{t-1}$, if no consumer other than u has issued an interest for C. We partition the output space $\Omega = \mathit{Range}(\mathcal{Q}_0^t)\cup \mathit{Range}(\mathcal{Q}_1^t)$ into Ω_1 and Ω_2, for all t and C, as follows:

- $\Omega_1 = Range(\mathcal{Q}_0^t) \setminus Range(\mathcal{Q}_1^t)$: If no consumer downstream from G other than u has issued an interest for C, then the first interest issued will result in a cache miss (in S_0, u must not have issued an interest for C yet either). However, this cannot occur in S_1, as u would have already requested C and it would be in G's collaborative cache. Therefore, $\nexists r_1$ such that $Q_0^t(C, r_0) = Q_1^t(C, r_1)$.
- $\Omega_2 = Range(\mathcal{Q}_0^t) \cap Range(\mathcal{Q}_1^t)$: Either some consumer other than u has requested C in S_0, or we are in S_1 so u has issued an interest for C (but may not be the only consumer to have done so). Either way, the output will be t cache hits: $\{1\}^t$. Thus, $Q_0^t(C, r_0) = Q_1^t(C, r_1)$.

Note that $\Omega_1 \cup \Omega_2 = \Omega$, as there are no possible outputs of $\mathcal{Q}_1^t$ that are not possible outputs of $\mathcal{Q}_0^t$ (whereas the converse is true).

A series of t ones is the only output $O \in \Omega_2$. Therefore, for all $O \in \Omega_2$, $\Pr[\mathcal{Q}_1^t = O] = 1$. For all $O \in \Omega_2$,

$$\begin{aligned}
\Pr[\mathcal{Q}_0^t = O] &= \Pr[\exists v \in U \setminus \{u\} \text{ s.t. } S_0(C, v) \geq 1] \\
&= \sum_{v=0}^{|r_0|-1} \Pr[S_0(C, v) \geq 1] = \sum_{v=0}^{|r_0|-1} (1 - \Pr[S_0(C, v) = 0]) \\
&= \sum_{v=0}^{|r_0|-1} \left(1 - \left(1 - \frac{1}{|\Gamma|}\right)^{m_{(v,0)}}\right) = \sum_{v=0}^{|r_0|-1} 1 - \sum_{v=0}^{|r_0|-1} \left(1 - \frac{1}{|\Gamma|}\right)^{m_{(v,0)}} \\
&= |r_0| - \sum_{v=0}^{|r_0|-1} \left(1 - \frac{1}{|\Gamma|}\right)^{m_{(v,0)}}
\end{aligned} \tag{2}$$

Substituting these values into clause 1 of Definition 1, we obtain

$$\forall O \in \Omega_2, \frac{\Pr[\mathcal{Q}_1^t = O]}{\Pr[\mathcal{Q}_0^t = O]} = \frac{1}{|r_0| - \sum\limits_{v=0}^{|r_0|-1} \left(1 - \frac{1}{|\Gamma|}\right)^{m_{(v,0)}}} \tag{3}$$

To circumvent the issue of division by zero, we assume there is at least a single piece of content in the network and each consumer downstream from the collaborating router cluster has issued at least one interest (for at least one piece of content). We then derive the value of ϵ as defined in Definition 1:

$$\begin{aligned}
&|\mathit{\Gamma}| \geq 1, m_{(v,0)} \geq 1 \ \forall v \in U \setminus \{u\} \Rightarrow \forall v \in U \setminus \{u\},\ 1 > \left(1 - \frac{1}{|\mathit{\Gamma}|}\right)^{m_{(v,0)}} \\
&\Rightarrow \sum_{v=0}^{|r_0|-1} 1 > \sum_{v=0}^{|r_0|-1} \left(1 - \frac{1}{|\mathit{\Gamma}|}\right)^{m_{(v,0)}} \\
&\Rightarrow |r_0| > \sum_{v=0}^{|r_0|-1} \left(1 - \frac{1}{|\mathit{\Gamma}|}\right)^{m_{(v,0)}} \Rightarrow |r_0| - \sum_{v=0}^{|r_0|-1} \left(1 - \frac{1}{|\mathit{\Gamma}|}\right)^{m_{(v,0)}} \geq 1 \\
&\Rightarrow \frac{1}{|r_0| - \sum\limits_{v=0}^{|r_0|-1} \left(1 - \frac{1}{|\mathit{\Gamma}|}\right)^{m_{(v,0)}}} = \frac{\Pr[\mathcal{Q}_1^t = O]}{\Pr[\mathcal{Q}_0^t = O]} \leq 1
\end{aligned} \tag{4}$$

Having determined an upper bound on $\frac{\Pr[\mathcal{Q}_1^t = O]}{\Pr[\mathcal{Q}_0^t = O]}$, we now calculate a lower bound on the same value in terms of ϵ.

$$\frac{1}{e^{\epsilon}} \leq \frac{1}{|r_0| - \sum\limits_{v=0}^{|r_0|-1} \left(1 - \frac{1}{|\mathit{\Gamma}|}\right)^{m_{(v,0)}}} = \frac{\Pr[\mathcal{Q}_1^t = O]}{\Pr[\mathcal{Q}_0^t = O]} \Rightarrow e^{\epsilon} \geq |r_0| - \sum_{v=0}^{|r_0|-1} \left(1 - \frac{1}{|\mathit{\Gamma}|}\right)^{m_{(v,0)}} \tag{5}$$

Under the reasonable assumption that the total number of requests made by any given consumer downstream from a collaborating router cluster will grow faster than the amount of content in the network, we find $\lim_{m_{(v,0)} \to \infty} \left(1 - \frac{1}{|\mathit{\Gamma}|}\right)^{m_{(v,0)}} = 0$, meaning this value approaches 0 as the number of interests issued in a given network state increase- a natural consequence of typical network traffic. Substituting this limit into the RHS of our inequality to allow us to calculate a concrete value for ϵ, we arrive at:

$$e^{\epsilon} \geq |r_0| - \sum_{v=0}^{|r_0|-1} \Rightarrow e^{\epsilon} \geq |r_0| \Rightarrow \ln e^{\epsilon} \geq \ \ln |r_0| \Rightarrow \epsilon \geq \ln |r_0| \tag{6}$$

Combining the upper and lower bounds computed on $\frac{\Pr[\mathcal{Q}_1^t = O]}{\Pr[\mathcal{Q}_0^t = O]}$, we conclude:

$$e^{-\ln |r_0|} \leq \frac{\Pr[\mathcal{Q}_1^t = O]}{\Pr[\mathcal{Q}_0^t = O]} \leq 1 \leq e^{\ln |r_0|} \therefore \epsilon = \ln |r_0| \tag{7}$$

We now derive the value of δ as defined in clause 2 of Definition 1. A zero followed by $t-1$ ones is the only output $O \in \mathit{\Omega}_1$. If $O \in \mathit{\Omega}_1$,

$$\begin{aligned}
&\delta = \Pr[\mathcal{Q}_0^t = O] + \Pr[\mathcal{Q}_1^t = O] = \Pr[\mathcal{Q}_0^t = O] + 0 = \Pr[S_0(C, v) = 0], \forall v \in U \setminus \{u\} \\
&= \prod_{v=0}^{|r_0|-1} \Pr[S_0(C, v) = 0] = \prod_{v=0}^{|r_0|-1} \left(1 - \frac{1}{|\mathit{\Gamma}|}\right)^{m_{(v,0)}} \\
&= \prod_{v=\ 0}^{|r_0|-1} \left(1 - m_{(v,0)} \cdot \frac{1}{|\mathit{\Gamma}|}\right) = \left(1 - \frac{1}{|\mathit{\Gamma}|}\right)^{\sum\limits_{v=0}^{|r_0|-1} m_{(v,0)}}
\end{aligned} \tag{8}$$

Finally, we determine an appropriate value for k as defined in Definition 2. As previously stated, S_0 and S_1 differ only in requests made for content object C. Therefore, $M = \{C\}$, where M is defined as in Definition 2. The only stipulation which must be satisfied by the chosen value of k is therefore that $0 < S_1(C, u) \leq k$. $S_1(C, u)$ is defined to be the cumulative number of requests made by router u for content C in state S_1. The maximum possible number of such requests is the cumulative number of requests for all content made by all consumers collectively in state S_1 (in the case where only consumer u issues any interests and every one of those interests is for content C). Therefore, we let $k = \sum_{v=0}^{|U|-1} m_{(v,0)}$.

Though the values of ϵ and δ we have proven may at first appear too complex to be meaningful, it is illustrative to consider them in the context of a network as time (and with it network traffic) progresses. The value of ϵ we have found grows logarithmically with respect to the number of consumers downstream from a router cluster. Complementing the slow logarithmic growth of ϵ is the observation that, compared to the amount of content and requests in a network, the number of consumers attached to a router cluster could reasonably be expected to grow quite slowly (or even remain relatively stagnant). We can conclude something even more concrete about the value of δ in this context. Assuming, as before, that the number of requests issued by consumers in a network grows at a rate faster than the number of unique content objects in the network, we see that:

$$\lim_{m_{(v,0)} \to \infty} \delta = \lim_{m_{(v,0)} \to \infty} \left(1 - \frac{1}{|\Gamma|}\right)^{\sum\limits_{v=0}^{|r_0|-1} m_{(v,0)}} = 0$$

Note that the binary representation of the output of $Q_i(C)$ deliberately abstracts away any details of the *delay* associated with interest satisfaction beyond the granularity of a cache hit or miss. Any attack which leverages a timing side-channel to determine the precise router within a cluster at which content is stored, even if successful, reveals no more information about individual consumers' requests than the knowledge that the content is cached *anywhere* in the cluster. Because the router at which content is to be cached is chosen uniformly at random, knowledge of the content in a particular router's cache in no way leaks any information about which specific consumer downstream from the cluster requested that content.

4.3 Quantifying Utility

Definition 3 ***Utility*** *[2]. Let $\mathcal{H}(\rho)$ denote the random variable describing the distribution of the number of cache hits depending on the total number of requests ρ ($\rho \geq 1$). The utility function $u : \mathbb{N} \to \mathbb{R}+$ of a cache management scheme is defined as: $u(\rho) = \frac{1}{\rho}\mathbb{E}(\mathcal{H}(\rho))$*

Intuitively, we define utility as the expected number of cache hits as a fraction of total interests issued. Using notation and assumptions from Sect. 4.2, let G denote the set of all clusters in the network (cluster count $n = |G|$) and $r_{g,0}$ denote r_0 for a given cluster $g \in G$.

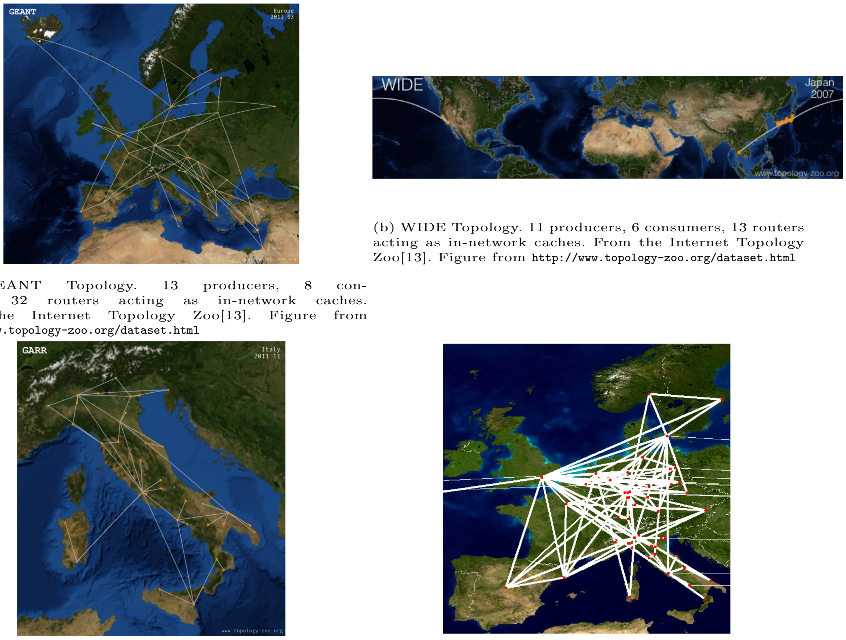
Theorem 2. *The utility $u(\rho) = \frac{1}{\rho}\mathbb{E}(\mathcal{H}(\rho))$ of Collaborative-Caching is:*

$$\frac{1}{\rho}\cdot\frac{\left(\sum_{g=0}^{n}\left(|r_{g,0}|-\sum_{v=0}^{|r_{g,0}|-1}\left(1-\frac{1}{|\Gamma|}\right)^{m_{(v,0)}}\right)\right)}{n}$$

Proof. If we consider the scale of a single collaborating router cluster, intuitively, the probability of a cache hit for a given interest is directly related to the probability that any consumer (including the one that issued the interest in question) downstream from the cluster has already requested the same content for which the interest was issued. Using the same notation as in our proof of Theorem 4.1, and again making the assumption that the content requested in consumers' interests is uniformly distributed, we denote this probability as the following:

$$\Pr[\exists v \in U \text{ s.t. } S_0(C,v) \geq 1] = \sum_{v=0}^{|r_0|-1}\left(1-\left(1-\frac{1}{|\Gamma|}\right)^{m_{(v,0)}}\right) = |r_0| - \sum_{v=0}^{|r_0|-1}\left(1-\frac{1}{|\Gamma|}\right)^{m_{(v,0)}} \tag{9}$$

for some network state S_0. Now consider the wider scope of all clusters in the network, letting $n = |G|$ denote the total number of clusters, where G is the



(a) GEANT Topology. 13 producers, 8 consumers, 32 routers acting as in-network caches. From the Internet Topology Zoo[13]. Figure from `http://www.topology-zoo.org/dataset.html`

(b) WIDE Topology. 11 producers, 6 consumers, 13 routers acting as in-network caches. From the Internet Topology Zoo[13]. Figure from `http://www.topology-zoo.org/dataset.html`

(c) GARR Topology. 13 producers, 21 consumers, 27 routers acting as in-network caches. From the Internet Topology Zoo[13]. Figure from `http://www.topology-zoo.org/dataset.html`

(d) TISCALI topology. 44 producers, 36 consumers, 160 routers acting as in-network caches. Parsed from the Rocketfuel dataset[24]. Figure from `https://research.cs.washington.edu/networking/rocketfuel/interactive/`

Fig. 2. Network topologies used in simulations.

set of all clusters in the network, and letting $r_{g,0}$ denote r_0 for a given cluster $g \in G$. Averaging the above probability over all clusters yields, as the value of $\mathbb{E}(\mathcal{H}(\rho))$:

$$\frac{\sum_{g=0}^{n-1}(|r_{g,0}| - \sum_{v=0}^{|r_{g,0}|-1}\left(1-\frac{1}{|\Gamma|}\right)^{m_{(v,0)}})}{n}$$

which we then multiply by $\frac{1}{\rho}$ to derive the expected cache hit ratio in Theorem 2.

5 Simulation Results

Having defined *Collaborative Caching*'s utility as a function of network traffic and topology, we now establish the practical implications of those bounds in a variety of simulated environments, with the intent of precisely quantifying the utility penalty one might expect to suffer in exchange for the proven privacy guarantees established in Sect. 4.2 and evaluating the impact of cache eviction on utility. Of particular interest is the comparative performance of multicast *Hash Routing* [21], a scheme expressly designed for improving ICN caching performance and which, when benchmarked against *Collaborative Caching*, should produce telling results regarding the trade-off between privacy and utility. Using the Icarus ICN caching performance simulation framework [22], experiments encompassing a variety of network topologies (detailed in Fig. 2), traffic characteristics, and caching schemes were performed. Constant factors in all experiments included the existence of $3x10^5$ unique content objects, $3x10^5$ "warmup" requests (issued prior to the beginning of performance measurements, to populate in-network caches), $6x10^5$ measured requests (used to compute results), an aggregate request rate of 1 per second, uniform content distribution amongst producers, and uniform cache space allocation amongst all in-network caches. Each experiment was parameterized by a unique combination of: cache eviction policy $p \in \{$Least Recently Used (LRU), Practical/In-Cache Least Frequently Used (LFU)$\}$, traffic including requests characterized by a Zipf distribution with coefficient $\alpha \in \{0.6, 0.8, 1.0\}$, total network cache size $n \in \{0.002, 0.008, 0.02\}$ as a fraction of content objects in the network, topology $t \in \{$GEANT, WIDE, GARR, TISCALI$\}$, and caching strategy $s \in \{$No Caching, Leave Copy Everywhere [11], Cache Less For More [3], ProbCache [20], Leave Copy Down [15], Random Choice [22], Random Bernoulli [22], Multicast Hash Routing [21], Collaborative Caching$\}$. For all topologies, "unclustered" variants of multicast *Hash Routing* and *Collaborative Caching* (wherein all cache nodes form one large cluster, implying a total cluster count of 1) were tested, whereas cluster counts of 2, 4, and 8 were also used in experiments involving "clustered" variants of multicast *Hash Routing* and *Collaborative Caching* on smaller topologies (WIDE and GARR), as opposed to cluster counts of 5, 10, and 20 for those same experiments on larger topologies (TISCALI and GEANT). Exhaustive simulations were conducted, including all possible experiments parameterized by each element of the

Table 1. Comparing the interest satisfaction latency and cache hit ratio observed in experiments pitting *Collaborative Caching* against a variety of other caching schemes using several network topologies. Zipf coefficient $\alpha = 0.8$. Cache size as a fraction of total content objects in network = 0.008. Reported values over ten trials per experiment indicated as "<mean> ± <error>", where (mean − error, mean + error) denotes a confidence interval of 99%. Data grouped into columns by cache eviction policy (Least Recently Used (LRU) vs. Practical/In-Cache Least Frequently Used (LFU)) and evaluated performance metric (cache hit ratio vs. interest satisfaction latency in milliseconds).

Caching	Topology	LRU Cache Hit Ratio	LRU Latency (ms)	LFU Cache Hit Ratio	LFU Latency (ms)
No Caching	GEANT	0.0000 ± 0.0000	87.1733 ± 0.0411	0.0000 ± 0.0000	87.2018 ± 0.0449
	WIDE	0.0000 ± 0.0000	78.2623 ± 0.0388	0.0000 ± 0.0000	78.2523 ± 0.0578
	GARR	0.0000 ± 0.0000	81.5429 ± 0.0243	0.0000 ± 0.0000	81.5514 ± 0.0261
	TISCALI	0.0000 ± 0.0000	91.9094 ± 0.0418	0.0000 ± 0.0000	91.9487 ± 0.0360
Leave Copy Everywhere	GEANT	0.1340 ± 0.0013	77.2756 ± 0.0875	0.2490 ± 0.0009	68.0778 ± 0.0531
	WIDE	0.1131 ± 0.0025	70.0952 ± 0.1585	0.2175 ± 0.0015	62.4335 ± 0.1063
	GARR	0.0877 ± 0.0023	75.1306 ± 0.1537	0.1931 ± 0.0014	67.2137 ± 0.1026
	TISCALI	0.0629 ± 0.0029	87.2501 ± 0.2061	0.1592 ± 0.0014	79.3937 ± 0.0980
Cache Less for More	GEANT	0.0976 ± 0.0012	79.3849 ± 0.0976	0.1455 ± 0.0021	75.4987 ± 0.1557
	WIDE	0.1175 ± 0.0012	69.6373 ± 0.0795	0.1851 ± 0.0016	64.7108 ± 0.1153
	GARR	0.1257 ± 0.0016	72.0542 ± 0.1068	0.1748 ± 0.0017	68.5384 ± 0.1129
	TISCALI	0.0810 ± 0.0015	85.2013 ± 0.1149	0.1142 ± 0.0031	82.9270 ± 0.2286
ProbCache	GEANT	0.1901 ± 0.0012	73.1032 ± 0.0810	0.2097 ± 0.0009	70.8426 ± 0.0762
	WIDE	0.1600 ± 0.0021	66.7402 ± 0.1302	0.2065 ± 0.0015	63.2143 ± 0.0988
	GARR	0.1346 ± 0.0012	71.8143 ± 0.0769	0.1787 ± 0.0010	68.2259 ± 0.0675
	TISCALI	0.0966 ± 0.0023	84.7233 ± 0.1597	0.1254 ± 0.0019	81.7880 ± 0.1281
Leave Copy Down	GEANT	0.1901 ± 0.0006	72.2230 ± 0.0482	0.2321 ± 0.0010	69.3720 ± 0.0750
	WIDE	0.1552 ± 0.0012	66.9467 ± 0.0728	0.2179 ± 0.0018	62.4464 ± 0.1200
	GARR	0.1420 ± 0.0012	70.8519 ± 0.0873	0.1865 ± 0.0020	67.7530 ± 0.1326
	TISCALI	0.1111 ± 0.0011	82.9278 ± 0.0867	0.1492 ± 0.0026	80.4490 ± 0.1715
Random Choice	GEANT	0.1714 ± 0.0008	74.2851 ± 0.0523	0.2451 ± 0.0012	68.3600 ± 0.0832
	WIDE	0.1396 ± 0.0021	68.1645 ± 0.1348	0.2176 ± 0.0014	62.4283 ± 0.0900
	GARR	0.1136 ± 0.0013	73.1932 ± 0.0904	0.1896 ± 0.0017	67.4475 ± 0.1115
	TISCALI	0.0848 ± 0.0030	85.4584 ± 0.2096	0.1544 ± 0.0018	79.7452 ± 0.1239
Random Bernoulli	GEANT	0.1691 ± 0.0008	74.5642 ± 0.0509	0.2462 ± 0.0007	68.2988 ± 0.0494
	WIDE	0.1408 ± 0.0022	68.0703 ± 0.1325	0.2171 ± 0.0010	62.4687 ± 0.0634
	GARR	0.1136 ± 0.0022	73.2183 ± 0.1399	0.1908 ± 0.0013	67.3758 ± 0.0870
	TISCALI	0.0840 ± 0.0034	85.6602 ± 0.2316	0.1573 ± 0.0016	79.5238 ± 0.1101
Multicast Hash Routing	GEANT (1 Cluster)	0.2024 ± 0.0004	77.1330 ± 0.0360	0.2980 ± 0.0007	69.9174 ± 0.0533
	GEANT (5 Clusters)	0.1498 ± 0.0011	81.0665 ± 0.2781	0.2614 ± 0.0024	71.8809 ± 0.5363
	GEANT (10 Clusters)	0.1445 ± 0.0031	80.3193 ± 0.6959	0.2581 ± 0.0031	71.4754 ± 0.5042
	GEANT (20 Clusters)	0.1327 ± 0.0030	80.3586 ± 0.4330	0.2520 ± 0.0034	70.5184 ± 0.4933
	WIDE (1 Cluster)	0.2022 ± 0.0003	69.4218 ± 0.0231	0.2992 ± 0.0009	62.5137 ± 0.0660
	WIDE (2 Clusters)	0.1772 ± 0.0027	70.7592 ± 0.5882	0.2753 ± 0.0017	63.9659 ± 0.5566
	WIDE (4 Clusters)	0.1529 ± 0.0037	71.8960 ± 0.8385	0.2594 ± 0.0058	64.2794 ± 0.5646
	WIDE (8 Clusters)	0.1412 ± 0.0039	70.5638 ± 0.5580	0.2391 ± 0.0035	62.7147 ± 0.4347
	GARR (1 Cluster)	0.2022 ± 0.0005	72.1170 ± 0.0419	0.2985 ± 0.0007	65.1865 ± 0.0504
	GARR (2 Clusters)	0.1678 ± 0.0016	76.0041 ± 0.1719	0.2834 ± 0.0014	67.3761 ± 0.1233
	GARR (4 Clusters)	0.1540 ± 0.0010	75.9113 ± 0.2279	0.2633 ± 0.0010	67.5829 ± 0.1285
	GARR (8 Clusters)	0.1435 ± 0.0036	76.0570 ± 0.4793	0.2488 ± 0.0022	67.9702 ± 0.4740
	TISCALI (1 Cluster)	0.2023 ± 0.0005	85.9880 ± 0.0348	0.2986 ± 0.0009	78.6860 ± 0.0671
	TISCALI (5 Clusters)	0.1447 ± 0.0027	91.9963 ± 0.3347	0.2515 ± 0.0033	83.1105 ± 0.2408
	TISCALI (10 Clusters)	0.1361 ± 0.0043	91.6054 ± 0.4447	0.2393 ± 0.0023	82.6559 ± 0.3442
	TISCALI (20 Clusters)	0.1221 ± 0.0043	92.1700 ± 1.1268	0.2261 ± 0.0049	83.7201 ± 0.4924
Collaborative Caching	GEANT (1 Cluster)	0.2019 ± 0.0003	77.1064 ± 0.0229	0.2983 ± 0.0006	69.8392 ± 0.0500
	GEANT (5 Clusters)	0.1501 ± 0.0021	81.1400 ± 0.4658	0.2619 ± 0.0014	71.9911 ± 0.4231
	GEANT (10 Clusters)	0.1424 ± 0.0022	80.4245 ± 0.3510	0.2557 ± 0.0030	71.6417 ± 0.4440
	GEANT (20 Clusters)	0.1357 ± 0.0028	80.2378 ± 0.4199	0.2543 ± 0.0030	70.2752 ± 0.3869
	WIDE (1 Cluster)	0.2021 ± 0.0005	69.3865 ± 0.0365	0.2989 ± 0.0008	62.4855 ± 0.0626
	WIDE (2 Clusters)	0.1755 ± 0.0033	71.0600 ± 0.7987	0.2756 ± 0.0022	63.7720 ± 0.6622
	WIDE (4 Clusters)	0.1593 ± 0.0032	71.5101 ± 0.5026	0.2565 ± 0.0077	64.2052 ± 0.8650
	WIDE (8 Clusters)	0.1390 ± 0.0048	71.0124 ± 0.6516	0.2449 ± 0.0068	63.1633 ± 0.8622
	GARR (1 Cluster)	0.2019 ± 0.0004	72.1890 ± 0.0332	0.2985 ± 0.0006	65.2583 ± 0.0396
	GARR (2 Clusters)	0.1695 ± 0.0013	75.9091 ± 0.1358	0.2829 ± 0.0015	67.5114 ± 0.1462
	GARR (4 Clusters)	0.1544 ± 0.0016	75.7915 ± 0.3114	0.2627 ± 0.0010	67.5725 ± 0.1946
	GARR (8 Clusters)	0.1440 ± 0.0030	76.1438 ± 0.4875	0.2504 ± 0.0023	67.9309 ± 0.3139
	TISCALI (1 Cluster)	0.2003 ± 0.0003	86.4305 ± 0.0235	0.2973 ± 0.0009	79.0758 ± 0.0690
	TISCALI (5 Clusters)	0.1444 ± 0.0009	92.0837 ± 0.2891	0.2501 ± 0.0016	83.2046 ± 0.3384
	TISCALI (10 Clusters)	0.1331 ± 0.0031	92.3790 ± 0.3812	0.2384 ± 0.0025	83.4196 ± 0.6007
	TISCALI (20 Clusters)	0.1216 ± 0.0036	92.4759 ± 0.2060	0.2230 ± 0.0049	84.5297 ± 0.8395

Cartesian product of all aforementioned parameter sets (10 trials for each of 1,080 unique experiments).

The data produced by the experiments with a Zipf coefficient of 0.8 and total network cache size of 0.008 (as a fraction of content objects in the network) proved to be representative of trends in the larger collected results as a whole, and is therefore provided in Table 1 as a focused subset thereof. Predictably, the performance of all schemes (with the exception of "No Caching") improved as aggregate cache size and the Zipf coefficient α (indicating the relative similarity/overlap of content requests) increased. *Collaborative Caching* consistently performed on par with *Hash Routing* regardless of cluster count, in some cases out-performing it relative to both latency and cache hit ratio metrics, and occasionally trailing *Hash Routing*'s performance by a very thin margin. Unclustered *Collaborative Caching* and unclustered *Hash Routing* achieved notably higher cache hit ratios than other schemes for each topology, and were often among the schemes with the lowest reported interest satisfaction latencies, as well. Interestingly, as cluster count decreased (and cluster size consequently increased), both *Collaborative Caching* and *Hash Routing* performed more favorably (lower latencies and higher cache hit ratios).

This trend is likely the result of the focus of our chosen simulation framework (namely, the measurement of caching performance). Our scheme has the potential to increase utility and privacy simultaneously. As cluster size increases, the likelihood that a given interest intercepted by the cluster corresponds to content cached in the cluster must monotonically increase, regardless of the distribution of content and interests. Also, the number of connected consumers must monotonically increase, increasing the size of the anonymity set of which downstream consumers are a part. The downside of increased cluster size is the overhead incurred by coordination and communication within the cluster, and simulating the resulting link saturation and congestion is not a problem Icarus claims to accurately emulate. We supplement these empirical observations with a theoretical calculation of *Collaborative Caching*'s utility as a function of network characteristics in Theorem 2.

6 Conclusions

We set out to demonstrate a caching scheme for ICN which would provide provable privacy guarantees and attack vector resilience for network consumers with negligible performance degradation. We have shown that, in a variegated pool of simulated environments, the interest satisfaction latencies and cache hit ratios afforded by our caching scheme are comparable to, and occasionally better than, those observed when schemes solely designed for improving cache utility are used. However, unlike those alternative methods, *Collaborative Caching* is able to accomplish this whilst preserving consumer privacy.

References

1. Abani, N., Braun, T., Gerla, M.: Betweenness centrality and cache privacy in information-centric networks. In: Proceedings of the 5th ACM Conference on Information-Centric Networking, ICN: 518, pp. 106–116. Association for Computing Machinery, New York (2018). https://doi.org/10.1145/3267955.3267964
2. Acs, G., Conti, M., Gasti, P., Ghali, C., Tsudik, G., Wood, C.A.: Privacy-aware caching in information-centric networking. IEEE Trans. Dependable Secure Comput. **16**(2), 313–328 (2019). https://doi.org/10.1109/TDSC.2017.2679711
3. Chai, W.K., He, D., Psaras, I., Pavlou, G.: Cache "less for more" in information-centric networks (extended version). Comput. Commun. **36**(7), 758–770 (2013). https://doi.org/10.1016/j.comcom.2013.01.007, https://doi.org/10.1016/j.comcom.2013.01.007
4. DiBenedetto, S., Gasti, P., Tsudik, G., Uzun, E.: Andana: anonymous named data networking application. In: 19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, 5–8 February, 2012 (2012), https://www.ndss-symposium.org/ndss2012/andana-anonymous-named-data-networking-application
5. Fotiou, N., Nikander, P., Trossen, D., Polyzos, G.C.: Developing information networking further: from PSIRP to PURSUIT. In: Tomkos, I., Bouras, C.J., Ellinas, G., Demestichas, P., Sinha, P. (eds.) BROADNETS 2010. LNICST, vol. 66, pp. 1–13. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30376-0_1
6. Gasti, P., Tsudik, G.: Content-centric and named-data networking security: the good, the bad and the rest. In: 2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), pp. 1–6, June 2018. https://doi.org/10.1109/LANMAN.2018.8475052
7. Ghali, C., Tsudik, G., Wood, C.A.: When encryption is not enough: privacy attacks in content-centric networking. In: Proceedings of the 4th ACM Conference on Information-Centric Networking, ICN 2017, pp. 1–10. ACM, New York (2017). https://doi.org/10.1145/3125719.3125723, http://doi.acm.org/10.1145/3125719.3125723
8. Gotz, M., Machanavajjhala, A., Wang, G., Xiao, X., Gehrke, J.: Publishing search logs–a comparative study of privacy guarantees. IEEE Trans. Knowl. Data Eng. **24**(3), 520–532 (2012)
9. ISO: ISO 80000–2: Quantities and units – Part 2: Mathematical signs and symbols to be used in the natural sciences and technology. International Organization for Standardization, Geneva, Switzerland, December 2009. https://www.iso.org/standard/31887.html
10. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L.: Networking named content. In: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, CoNEXT Õ 2009, pp. 1–12. Association for Computing Machinery, New York (2009). https://doi.org/10.1145/1658939.1658941
11. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L.: Networking named content. In: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, CoNEXT 2009, pp. 1–12. Association for Computing Machinery, New York (2009). https://doi.org/10.1145/1658939.1658941

12. Kaufman, L., Rousseeuw, P.J.: Partitioning Around Medoids (Program PAM), chap. 2, pp. 68–125. John Wiley & Sons, Ltd. (2008). https://doi.org/10.1002/9780470316801.ch2, https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470316801.ch2
13. Knight, S., Nguyen, H.X., Falkner, N., Bowden, R., Roughan, M.: The internet topology zoo. IEEE J. Sel. Areas Commun. **29**(9), 1765–1775 (2011)
14. Kumar, N., Aleem, A., Singh, A.K., Srivastava, S.: NBP: namespace-based privacy to counter timing-based attack in named data networking. J. Network Comput. Appl. **144**, 155–170 (2019). https://doi.org/10.1016/j.jnca.2019.07.004, http://www.sciencedirect.com/science/article/pii/S1084804519302280
15. Laoutaris, N., Che, H., Stavrakakis, I.: The LCD interconnection of LRU caches and its analysis. Perform. Eval. **63**(7), 609–634 (2006). https://doi.org/10.1016/j.peva.2005.05.003
16. Machanavajjhala, A., Kifer, D., Abowd, J., Gehrke, J., Vilhuber, L.: Privacy: theory meets practice on the map. In: 2008 IEEE 24th International Conference on Data Engineering, pp. 277–286 (2008)
17. Mohaisen, A., Mekky, H., Zhang, X., Xie, H., Kim, Y.: Timing attacks on access privacy in information centric networks and countermeasures. IEEE Trans. Dependable Secure Comput. **12**(6), 675–687 (2015)
18. Mohaisen, A., Zhang, X., Schuchard, M., Xie, H., Kim, Y.: Protecting access privacy of cached contents in information centric networks. In: Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, ASIA CCS: 513, pp. 173–178. Association for Computing Machinery, New York (2013). https://doi.org/10.1145/2484313.2484335, https://doi.org/10.1145/2484313.2484335
19. NDN Project Homepage (2020). https://named-data.net/. Accessed 4 June 2020
20. Psaras, I., Chai, W.K., Pavlou, G.: Probabilistic in-network caching for information-centric networks. In: Proceedings of the Second Edition of the ICN Workshop on Information-Centric Networking, ICN 2012, pp. 55–60. Association for Computing Machinery, New York (2012). https://doi.org/10.1145/2342488.2342501, https://doi.org/10.1145/2342488.2342501
21. Saino, L., Psaras, I., Pavlou, G.: Hash-routing schemes for information centric networking. In: Proceedings of the 3rd ACM SIGCOMM Workshop on Information-Centric Networking, ICN 2013, pp. 27–32. Association for Computing Machinery, New York (2013). https://doi.org/10.1145/2491224.2491232
22. Saino, L., Psaras, I., Pavlou, G.: Icarus: a caching simulator for information centric networking (ICN). In: Proceedings of the 7th International ICST Conference on Simulation Tools and Techniques (SIMUTOOLS 2014). ICST, ICST, Brussels, Belgium (2014)
23. Sourlas, V., Psaras, I., Saino, L., Pavlou, G.: Efficient hash-routing and domain clustering techniques for information-centric networks. Comput. Networks **103**, 67–83 (2016). https://doi.org/10.1016/j.comnet.2016.04.001, http://www.sciencedirect.com/science/article/pii/S1389128616300998
24. Spring, N., Mahajan, R., Wetherall, D.: Measuring ISP topologies with rocketfuel. SIGCOMM Comput. Commun. Rev. **32**(4), 133–145 (2002). https://doi.org/10.1145/964725.633039
25. Tsudik, G., Uzun, E., Wood, C.A.: Ac3n: anonymous communication in content-centric networking. In: 2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC), pp. 988–991, January 2016. https://doi.org/10.1109/CCNC.2016.7444924

26. Xue, K., et al.: A secure, efficient, and accountable edge-based access control framework for information centric networks. IEEE/ACM Trans. Networking **27**(3), 1220–1233 (2019)
27. Yao, L., Jiang, B., Deng, J., Obaidat, M.S.: LSTM-based detection for timing attacks in named data network. In: 2019 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2019)