Trace-Driven Analysis of ICN Caching Algorithms on Video-on-Demand Workloads

Yi Sun[⊗], Seyed K. Fayaz[†], Yang Guo[⊗], Vyas Sekar[†]
Yun Jin[◇], Mohamed Ali Kaafar[*], Steve Uhlig[◦]
[⊗] ICT/CAS, [†] CMU, [◇] PPTV, [*] NICTA, [◦] QMUL
{sunyi, guoyang}@ict.ac.cn, seyed@cmu.edu, vsekar@andrew.cmu.edu
yunjin@pptv.com, dali.kaafar@nicta.com.au, steve@eecs.qmul.ac.uk

ABSTRACT

Even though a key driver for Information-Centric Networking (ICN) has been the rise in Internet video traffic, there has been surprisingly little work on analyzing the interplay between ICN and video – which ICN caching strategies work well on video workloads and how ICN helps improve video-centric quality of experience (QoE). In this work, we bridge this disconnect with a trace-driven study using 196M video requests from over 16M users on a country-wide topology with 80K routers. We evaluate a broad space of content replacement (e.g., LRU, LFU, FIFO) and content placement (e.g., leave a copy everywhere, probabilistic) strategies over a range of cache sizes.

We highlight four key findings: (1) the best placement and replacement strategies depend on the cache size and vary across improvement metrics; that said, LFU+probabilistic caching [37] is a close-to-optimal strategy overall; (2) video workloads show considerable caching-related benefits (e.g., $\geq$ 10% traffic reduction) only with very large cache sizes ($\geq$ 100GB); (3) the improvement in video QoE is low ($\leq$ 12%) if the content provider already has a substantial geographical presence; and (4) caches in the middle and the edge of the network, requests from highly populated regions and without content servers, and requests for popular content contribute most to the overall ICN-induced improvements in video QoE.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed systems—*Distributed applications*

General Terms

Design, Management, Measurement, Performance

Keywords

Information-centric networking; caching; Internet video

1. INTRODUCTION

Many future Internet architectures incorporate key information-centric networking (ICN) concepts (e.g, [6, 12, 27, 31, 33]). This is rooted in the observation that Internet traffic consumption is largely *content driven*, as users largely care about "what" content they want as opposed to "where" the content is delivered from. This decoupling promises to simplify several aspects of networking today by eliminating complex optimizations required by CDNs, providing intrinsic trust embedded in the content itself rather than network sources, and enabling the network to handle larger traffic volumes.

This motivation for ICN is accompanied, and driven in large part, by the dramatic rise in Internet video traffic volumes over the last several years [3]. In particular, the promise of ubiquitous caching and simplified network traffic engineering that ICN architectures offer have a natural synergy with the quality expectations and bandwidth demands of Internet video workloads. In this context, there are two natural questions that arise with respect to the interaction between ICN and Internet video:

- *How do video workloads impact network-level improvements offered by different ICN caching strategies?*

  The canonical improvement metrics studied in the ICN literature include cache hit rate, origin server load reduction, and the reduction in the overall network footprint [23]. There is a very rich literature on different aspects of caching to optimize these metrics including work on *content placement* to decide which routers on a request path should cache the content (e.g., [18, 22, 36, 37, 43]) and *content replacement* to decide how to manage the cache storage when full (e.g., [9, 42]). We would like to understand the specific placement and replacement algorithms that work well for video workloads and what magnitude of improvement they can offer on video workloads.
- *How do ICN caching mechanisms impact key video-specific QoE metrics?*

  Unlike traditional Internet web workloads, Internet video introduces new QoE metrics that impact user engagement such as buffering ratio, startup latency, and average bitrate [21, 34]. Going beyond the aforementioned network-centric metrics, we would also like to understand how different ICN caching strategies impact these key video-specific QoE metrics.

Despite the tremendous interest in both ICN and Internet video, there has been little work on systematically answering these questions with actual video workloads and using real network topologies. Much of the ICN literature today has focused on generic Zipf-like workloads and on small-scale topologies [44,45,49]. Similarly, most of the Internet video literature focuses on traditional CDN-based delivery architectures [14, 21, 28, 34, 41]. Thus, there is a gap in our understanding of the interplay between the design space of ICN content placement and replacement strategies and video workloads and QoE considerations.

This paper bridges this gap using large-scale video on demand (VOD) request traces consisting of over 196 million video requests

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CoNEXT'14, December 2–5, 2014, Sydney, Australia.
Copyright 2014 ACM 978-1-4503-3279-8/14/12 ...$15.00.
http://dx.doi.org/10.1145/2674005.2675003.

from over 16 million users from the PPTV deployment in China [8]. We evaluate a combination of seven ICN content placement [18, 22, 36, 37, 43], five content replacement schemes [9, 42], and four different cache sizes (1GB, 10GB, 100GB, and 1TB). For each of these scenarios, we run trace-driven simulations on a country-wide router-level topology with around 80K routers. To the best of our knowledge, this is the largest (in terms of the topology and number of requests) and most comprehensive (in terms of the design space) trace-driven analysis of ICN caching strategies on video workloads.

Running such a large-scale workload on a large topology and a broad range of scenarios raises significant scalability challenges for existing ICN simulation frameworks [10]. We develop a custom request-level simulator called iCache for our analysis that provides almost 50× improvement over state-of-the-art ICN simulation platforms.

Using iCache, we answer the two high-level questions raised above in the context of a popular video content provider (PPTV). Our key observations are as follows:

- Across all strategies, we observe that only very large caches (i.e., 100GB or 1TB) provide considerable ($\geq 10\%$) reduction in the network traffic.
- For a provider like PPTV that already has a substantial server footprint, even with the largest cache size in our evaluations (1TB), the overall video QoE improvement is at most 12%.
- While the best combination of content placement and replacement strategies does depend on the cache size and the metric of interest, the combination of probabilistic content placement (*Prob*) and LFU content replacement (*LFU*) emerges as a near-optimal strategy across all scenarios.
- We also analyze *where*, *when*, and *what* contributes to the improvement and find that: (a) caches in the middle and access portions of the network contribute most to traffic reduction and QoE improvement; (b) requests from highly populated regions without content servers observe the highest QoE improvement; and (c) requests for popular content are more likely to contribute to overall ICN-induced QoE improvements.

Contributions and Roadmap: To summarize, the main contributions of our work are:

- Developing a scalable and extensible ICN simulator that provides up to 50× improvement over state-of-the-art platforms (Section 4);
- Analyzing the impact of video workloads on canonical ICN metrics such as traffic reduction, hit rate, and server load reduction over a broad combination of ICN caching strategies (Section 5);
- Understanding the improvements that ICN caching algorithms offer for video QoE metrics (Section 6);
- A spatiotemporal and request-level dissection of the benefits of the best ICN strategies with respect to network-centric and QoE-centric metrics (Section 7).

In the rest of the paper, we begin with related work in Section 2. We describe our datasets in Section 3. We discuss some outstanding issues in Section 8 before concluding in Section 9.

2. BACKGROUND AND RELATED WORK

In this section, we review the most relevant related work on ICN, ICN caching algorithms, ICN evaluation platforms, and measurements of Internet video. At a high level we find that while there is a rich literature on analyzing ICN and video workloads separately, there is little work on evaluating ICN caching mechanisms *at scale* on real video workload on a large topology. Our work bridges this gap.

ICN vision: The main ideas in ICN are decoupling content names from locations, name-based content routing, in-network caching, and the use of name-based binding for security. Major ICN efforts include TRIAD [25], DONA [33], 4WARD [12], CCN [30], NDN [31], and COMET [4]. Many of these envision content caches on routers to improve user experience (e.g., lower latency) and network performance (e.g., lower server load and congestion). However, these ICN architectures are not evaluated with respect to video workloads, which is the focus of our work.

Design space of in-network caching: In-network caching plays a key role in ICN architectures and has attracted a lot of research along two main dimensions: *content placement* and *content replacement*.

A *content placement* strategy decides where an object should be cached across routers on the request path (i.e., the sequence of routers from the content origin to the requesting client). We study seven specific content placement strategies: (1) leaving a copy everywhere (*LCE*) [19]; (2) leaving a copy only on the immediate downstream router when there is a hit on an upstream router (*LCD*) [36]; (3) leaving a copy on some randomly chosen router along the request path (*Rand*) [22]; (4) constant probability of caching on each router (*Prob*) [37]; (5) the probability of caching on a router is a function of its distance from the origin server and the shared storage capacity of the path (*PProb*) [43]; (6) using centrality-based measures where the router with the largest value of betweenness[1] will keep a copy (*Centrality*) [18]; and (7) a hybrid approach where the probability that a router caches an object is a function of both the router's betweenness and the content's popularity (*Cross*) [48].

A *content replacement* strategy captures how objects are evicted when the cache is full. We consider five such strategies: (1) least-recently used (*LRU*); (2) first-in first-out (*FIFO*); (3) least frequently used (*LFU*); (4) time to live (*TTL*) [42], where an object's time to live (TTL) on a router's cache depends on object's popularity and router's location; and (5) largest item first (*Size*) [9].

Notably, content placement and replacement are fundamentally different from strategies for content staging, which essentially prepopulate or *initialize* the caches. In other words, content placement and replacement strategies together determine the steady state behavior of the caches rather than the initial conditions that may only impact a few initial requests.

ICN evaluation: There are many ICN evaluation platforms such as ndnSIM [10], ccnSim [20], DEC [5], and Mini-CCNx [15]. Unfortunately, our experiences showed that these cannot handle the scale of the workload and the topology we study in this paper. Thus, we developed a custom request-level simulator called iCache. Prior works have also considered trace-driven evaluations of ICN caching (e.g., [23,49]). Our work extends them in three significant ways. First, our evaluation considers a much larger topology and request workload. Second, we consider a broader spectrum of placement and replacement strategies. Finally, we focus on video-specific aspects of ICN.

Video over ICN: Recent work has extended ICN to support video streaming [35, 38], optimized ICN-based streaming on mobile devices [26], and has suggested cooperative caching to improve video delivery [39]. Closer to our work, Fricker et al. [24] found that

[1]Betweenness is a measure of a node's centrality in a network. It is equal to the ratio of shortest paths from all vertices to all others that pass through that node.

Nodes	Edges	Avg. Degree	Diameter	Avg. Path Length
82,726	167,921	4.06	22	8.26

Table 1: Attributes of router-level network topology.

VOD workloads are cacheable at the edge of the network using synthetic traffic and a small, synthetic hierarchical topology. Similarly, Ando and Nakao [11] evaluated the performance of caching on a YouTube workload collected at a campus network gateway. Our analysis uses much larger traces and a real country-wide topology.

Internet video measurement: Several studies have analyzed the content popularity in video workloads (e.g., [13, 17, 29, 50, 51]) and suggest implications for caching in CDN and edge proxy architectures. However, they do not consider ICN-like ubiquitous caching. Recent studies analyze the impact of QoE metrics such as buffering, startup delay, and average playback bitrate on user engagement [21, 34]. There are also proposals to improve QoE by designing better adaptive streaming protocols (e.g., [32]) and intelligently using multiple CDNs [40, 41]. However, these focus on existing CDN-based delivery architectures while we focus on the interplay between ICN and video QoE.

3. DATASET DESCRIPTION

In this section, we describe how we collected the datasets used to answer our two motivating questions. These datasets, a router-level network topology and session-level statistics of video views, were collected from the operational platform of PPTV [8]. PPTV is a leading online video content provider in China. (While PPTV originally started as a P2P delivery platform, today it is a well-provisioned commercial content provider with traditional CDN-style servers and P2P.) PPTV has a user base of more than 227 million users, and its active user base is 49.7% of the total number of Internet users in China. The average viewing time per user per day is more than 2.5 hours, making it one of the most popular content portals in China [7].

3.1 Topology data

To collect a router-level network topology, we instrumented the PPTV client software with `traceroute` capabilities. After PPTV released this instrumented software, more than 150 million users (i.e., 66% of PPTV users) updated their software. Out of the users with the updated client software, we randomly selected 1.68 million users for conducting traceroute measurements. These clients issue traceroutes to other peers, the CDN servers deployed by PPTV, as well as specific IP prefixes associated with Chinese ISPs. To balance the tradeoff between measurement overhead and coverage, we used simple heuristics in choosing the destinations and the time to issue traceroutes. Some examples of such heuristics include avoiding duplicate requests to peers within previously covered IP prefixes and issuing traceroutes when the network is less congested (e.g., early morning).

The topology used in this paper is based on traceroute measurements collected over a 2-month period. Given the popularity and spread of PPTV clients in China, we have a unique high-coverage map of a country-wide network covering 89 ISPs, 6583 PoPs, and all the 31 provinces in mainland China. Table 1 summarizes the key characteristics of the network topology.

3.2 Video views data

We also collected a dataset of *video-on-demand* (VOD) requests from PPTV server logs. Each log record represents the summary statistics of a single video request, which is pushed to log collection

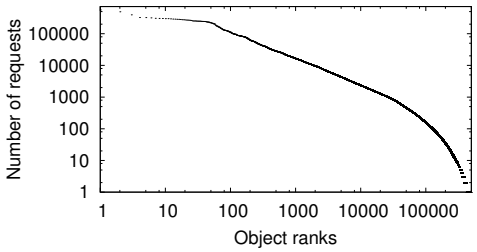


Figure 1: Object popularity distribution follows a Zipf-like distribution.

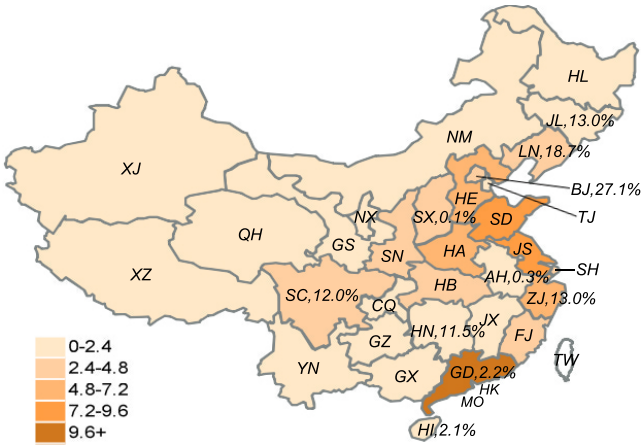


Figure 2: Distribution of incoming requests and number of requests served across different provinces in China. The "heatmap" scale represents the percentage of client requests originating in a given province, and the annotated text with each province is the percentage of requests served by PPTV servers in that province.

servers when the video session ends. There are four main categories of relevant entries:

1. *Client-specific information* such as a (unique) client ID, client IP, client ISP, client location city/province, client's access network type (e.g., ADSL, WiFi, 3G), and client terminal type (e.g., PC, Tablet, Smartphone).
2. *Server-specific information* such as the PPTV server IP and the ISP this server is located in.
3. *Session-level information* such as video name, start time of the request, view duration, and total bytes transferred during this video session.
4. *Video QoE information* such as:
   - Buffering ratio or the fraction of the session time spent on buffering $= \frac{BufferingTime}{TotalViewTime}$;
   - Average bitrate, which is $\frac{TotalBytesDownloaded}{TotalPlayTime}$; and
   - Join time or the delay between the request and video start $= StartTimeOfPlay - RequestTime$.

PPTV shared with us a (uniformly-at-random) sampled dataset over a two-week period with more than 196 million viewing instances involving more than 16 million users. (Due to business considerations, we cannot reveal the total number of sessions.) Overall, these users watched more than 500K unique videos with the total size of unique content views being 137TB. To put this in context, a large 1TB cache can hold $< 1\%$ of the entire content population. Figure 1 depicts the distribution of user access to video content: a Zipf-like distribution with the parameter $\alpha = 1.174$.

Figure 2 illustrates the spatial distribution of content access by PPTV users, showing the percentage of content access and the percentage of requests destined to PPTV content servers on a per-province basis. In Figure 2, darker colors indicate higher content requests, which matches very well with the Internet user population distribution in China.

4. SIMULATION SETUP

In this section, we discuss iCache, our simulation platform (written in C++) for large-scale ICN evaluations.[2] iCache is built with two design goals in mind: scalability and extensibility. First, in terms of scalability, iCache completes simulation of a very large network with tens of thousands of routers, tens of millions of ICN clients, and billions of content-view logs within a reasonable period of time (less than 5 hours). In contrast, existing platforms may take more than 250 hours for a similar simulation. Second, with respect to extensibility, in addition to supporting several existing caching and routing strategies, iCache provides APIs to plug in new algorithms.

We begin by documenting our experiences with existing simulation platforms that motivated us to develop iCache. Then, we describe the main design decisions underlying iCache and validate its scalability, fidelity, and extensibility.

4.1 Need for a new ICN simulation platform

There are several ICN simulation platforms currently in use including CCNx [1], ccnSim [20], DCE [5], mini-CCNx [15], and ndnSIM [10]. ndnSIM is generally considered to be the most scalable ICN platform; e.g., as reported by Afsanyev [10] et al., CCNx, ccnSim, DCE, and mini-CCNx can only support ICN simulations with a few hundred routers, and cannot be used for our large-scale evaluation with 80,000+ routers.

Thus, we originally considered using ndnSIM. Unfortunately, even ndnSIM was not sufficiently scalable for the large-scale trace-driven evaluation we needed to perform. For instance, a single trace-driven run with one content placement and one content replacement strategy took more than 125 hours on a Xeon 2.13GHz machine with 1TB of memory running Linux. This performance is clearly not sufficient for us to practically meet our goal of comprehensively evaluating the design space of content placement and replacement strategies with different cache sizes. With 7 content placement, 5 replacement strategies, and 4 distinct cache sizes, using ndnSIM would roughly take $125 \times 140 = 17,500$ CPU hours.

4.2 Design of iCache

In designing iCache, we make three key decisions:

- ndnSIM does a detailed packet-level simulation carefully capturing every content packet and content response. This leads to a significant number of events in the system and also increases memory consumption at high request rates. As a first step, we simplify the simulation by moving to a request-level rather than a packet-level simulation. One concern is that we may not be able to replicate fine-grained QoE metrics with a flow-level simulation. Fortunately, we use a data-driven extrapolation methodology for computing QoE metrics, as we discuss later in Section 6, that is amenable to a flow-level analysis. Second, ndnSIM provides a complete implementation of all ICN functionalities; e.g., checks for content-level security. These additional features, while relevant for other applications, significantly add to the per-request processing. We simplify the per-node operations and do not implement the content-integrity steps.
- To emulate complex content routing protocols such as OSPF extensions, ndnSIM effectively recomputes each routing table on each cache eviction event. This creates a large number of routing protocol messages and increases the processing overhead. However, for many common ICN routing strategies (e.g., shortest path to servers [23,47]) the routing table does not need to be updated. We, therefore, precompute the routing tables and use them throughout the simulation.
- Finally, rather than a node-centric software architecture, we move to a "tier"-based architecture where we group the same functional modules on different nodes into a single layer such as caching layer, forwarding layer, and routing layer as shown in Figure 3. Unlike other simulators, where each simulated ICN node maintains its own routing, caching, and forwarding tables, this grouping enables the use of global tables that are shared by all routers avoiding redundant entries. In turn, this reduces the memory consumption of the simulation. Furthermore, it improves the locality of memory accesses and thus minimizes I/O operations, and also improves the processing time of each simulated router.

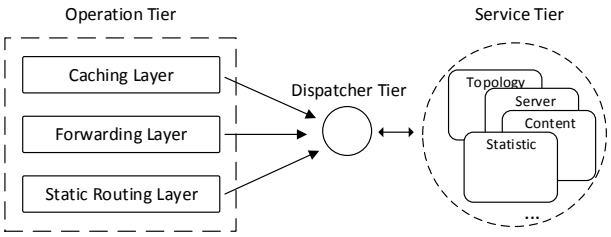


Figure 3: iCache architecture.

Figure 3 depicts the architecture of iCache including three functional tiers (i.e., the operation, dispatcher, and service tiers) as described next. The operation tier provides the core caching and routing functions. The dispatcher schedules the events in the simulation. Finally, the service tier provides a common set of functions such as queries on network topology and content properties as well as collecting statistics.

Here we focus on the operation tier. The *caching layer* implements the functionality of a content cache; i.e., basic operations of individual caches such as content placement/replacement. We have implemented seven content placement and five content replacement algorithms (see Section 5). The *forwarding layer* processes content request and reply messages and implements the basic forwarding functions such as receiving pending requests and forwarding the responses on the correct interfaces of each simulated router. The *routing layer* maintains global routing information of the network. We have currently implemented a name-based OSPF algorithm, which is the basic routing strategy in several ICN architectures [6, 31, 33].

We record several useful statistics such as the transmission path of each request and the traffic load on each server during the simulation. We process these offline using custom scripts to generate statistics such as cache hit rate, traffic reduction, server load reduction (Section 5), and video-related user experience metrics (Section 6).

4.3 Validation and evaluation

Next, we evaluate iCache on three key dimensions: fidelity, scalability, and extensibility.

[2]An open-source version of iCache will be available at the following URL: `http://fi.ict.ac.cn/firg.php?n=Member.Source`.

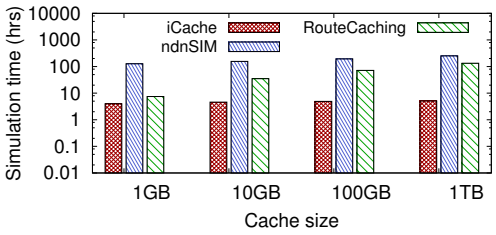


Figure 4: Simulation time of iCache and ndnSIM showing that iCache can provide 20–50× speedup. Note that the y-axis is on log scale.

Algorithm	LoC
LCE	17
LCD	17
Rand	23
Prob	29
PProb	38
Centrality	56
Cross	80

(a) Content placement

Algorithm	LoC
FIFO	108
LFU	233
LRU	140
TTL	184
Size	139

(b) Content replacement

Table 2: Lines of code for different algorithms in iCache.

Fidelity: First, we need to ensure that the optimizations we introduced in iCache do not introduce semantic bugs in the caching behavior. Thus, to evaluate the fidelity of iCache against ndnSim [10], we generated several topologies and content request traces. Due to the scalability limits of ndnSIM, we typically ran this for smaller topologies with less than 100 nodes. Then, we observed that the cached contents of corresponding routers and paths taken by the request exactly match across the two experiments (not shown). This being said, we acknowledge that iCache does not provide all the features of ndnSIM, especially in terms of advanced content routing protocols, security checks, and packet-level behaviors.

Scalability: Figure 4 shows the simulation time of a large-scale simulation on a machine with a Xeon 2.13GHz core and 1TB of memory running Linux. The simulation involved about 80K routers, 16M clients, 500K videos, and 196M content requests (see Section 3), and took $\leq$5 hours for different cache sizes. For comparison, the native ndnSIM needs 127.1 to 252.2 hours to finish the same scale of simulations. As a point of comparison, we also include a version of ndnSIM with routing pre-computation optimization described above. This precomputation does help significantly, as it drop the simulation time to 7.5-132.9 hours; but this is still almost 2× slower than iCache.

Extensibility: We evaluate the ease of adding new caching algorithms to iCache by counting the lines of codes for the 7 content placement and 5 content replacement algorithm modules written in iCache. Table 2 shows that across all the strategies we did not need more than about 230 lines of C++ code.

5. EFFECT OF VIDEO WORKLOADS ON ICN

In this section, we investigate the effect of the PPTV VOD workload on the performance of ICN and address the following questions:

- How does cache size impact the performance of ICN caching schemes in terms of network footprint reduction, cache hit rates, and server load reduction given video workloads?
- What is the best combination of content placement and replacement algorithms for different cache sizes and across different performance metrics?
- Is there a near-optimal or good enough strategy that is robust across this space of considerations?

5.1 Setup

First, we describe the simulation setup used in the rest of the paper. There are three types of nodes in the iCache platform: servers (i.e., content origins), consumers (i.e., content requesters), and routers. Servers correspond to the PPTV CDN servers. (Note that we do not consider the P2P mode in the evaluations and only focus on the CDN mode.) There are a total of 221 servers connected to the network from the same access points as in the real deployment of PPTV. We assume each server maintains all video contents. Consumers correspond to the PPTV clients observed in the dataset.

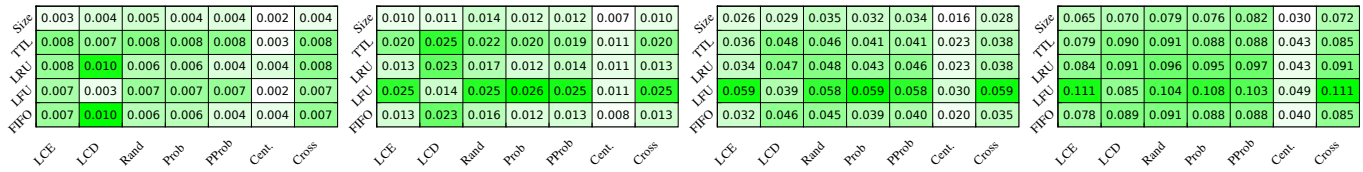
Each client is attached to the access router whose IP has the longest prefix in common with it. There are 82,726 routers each equipped with a cache of the same size. As in many previous works (e.g., [23,44,45,47]), we use the shortest paths to populate the routing tables of each router. Each content request is routed toward the closest origin server, and the request is "shortcut" if the content is found on an on-path router.

There are three key dimensions to explore here: (1) *cache size*, (2) *content placement strategy*, and (3) *content replacement strategy*. For (1), we consider four different cache sizes: 1GB, 10GB, 100GB, and 1TB. To put this in context, considering the total volume of different videos in our dataset is 137TB, these sizes of caches can approximately store 0.0007%, 0.007%, 0.07% and 0.7% of contents, respectively. As a reference point, we also consider a zero-cache scenario. For (2), we have implemented seven different content placement algorithms from related work: *LCE*, *LCD*, *Rand*, *Prob*, *PProb*, *Centrality*, and *Cross* [18, 22, 36, 37, 43, 48]. Finally, for (3), we consider five content replacement algorithms: *FIFO*, *LRU*, *LFU*, *TTL*, and *Size* [9, 42]. This space of parameters covers some of the relevant sizes for today's storage caches and represents main proposed caching strategies. In total, there are $4\times7\times5 = 140$ combinations of these parameters.

5.2 Results

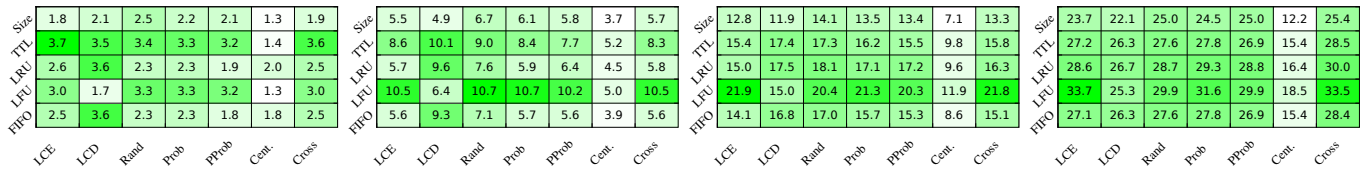
There are three main metrics used to quantify the performance of ICN [23] that we define below. Here the subscript r refers to a specific router and R is the number of routers. Similarly, the subscript q refers to a specific object request and V is the total number of video requests. The metrics are:

1. *Average cache hit ratio* across all routers defined as $HitRate = \frac{\sum_{r=1}^{R} \frac{Hit_r}{Hit_r + Misses_r}}{R}$, where Hit_r refers to the number of requests that router r was able to serve from its cache, and $Misses_r$ is the number of requests that it forwarded upstream due to cache miss.
2. *Total reduction in the network footprint* or the total number of byte-hops saved via in-network caching defined as: $TrafficReduction = \frac{\sum_{q=1}^{V} (HopsNoCache_q - HopsICN_q) \times Bytes_q}{\sum_{q=1}^{V} HopsNoCache_q \times Bytes_q}$, where $HopsNoCache_q$ refers to the number of hops the request would take to reach the nearest origin server, and $HopsICN_q$ is the number of hops the request took to reach



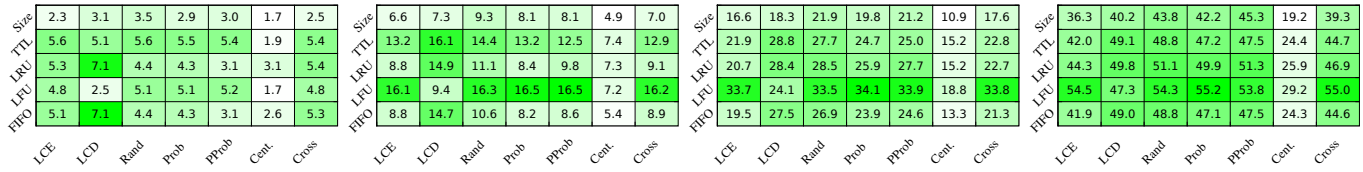
(a) 1GB (b) 10GB (c) 100GB (d) 1TB

Figure 5: Average cache hit ratio: The cache hit rates are generally low, and the best combination of placement and replacement for 1GB, 10GB, 100GB, and 1TB are *LCD +LRU*, *Prob +LFU*, *Prob +LFU*, and *Cross +LFU*, respectively.



(a) 1GB (b) 10GB (c) 100GB (d) 1TB

Figure 6: Total traffic reduction (%): The reduction is more than 20% only with cache > 100GB. The best combination of placement and replacement for 1GB, 10GB, 100GB, and 1TB are *LCE +TTL*, *Prob +LFU*, *LCE +LFU*, and *LCE +LFU*, respectively.



(a) 1GB (b) 10GB (c) 100GB (d) 1TB

Figure 7: Server load reduction (%): The reduction is substantial only when cache size is > 100GB. The best combination of placement and replacement for 1GB, 10GB, 100GB, and 1TB are *LCD +LRU*, *PProb +LFU*, *Prob +LFU*, and *Prob +LFU*, respectively.

the closest location containing a copy of the video content (i.e., either an in-network cache or the origin server).

3. *Reduction in load on the origin servers* defined as: $1 - \frac{NumCachedReq}{V}$, where $NumCachedReq$ is the number of requests that were served from *some* in-network cache.

Figures 5, 6, and 7 show the values of these three metrics across the space of 35 caching strategy combinations for four different cache sizes. We reiterate that such a comprehensive evaluation of the design and parameter space would not be possible without the scalability of iCache.

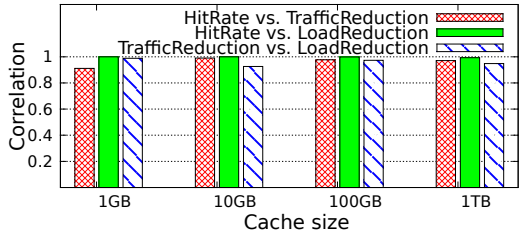
We make a few observations from these results. First, we see that the metrics improve as the cache size grows, and, because of the large traffic volume of videos, we observe significant benefits only with 100GB or 1TB cache sizes. Note that with a 1GB cache on each router, the average reduction of video traffic and cache hit rate of routers is less than 3.7% and 1.0%, respectively, across all caching strategies. However, when the cache size increases to 100GB, the above metrics improve to 21.9% and 5.9%, respectively. Second, we observe a sub-linear effect of adding more cache capacity or a natural "diminishing returns" property, as increasing the cache size by 10× provides less than 2× performance improvement, even with the best possible strategy. Third, we do observe that some of the more "intelligent" content placement-replacement combinations can provide substantial improvements over the vanilla *LCE +LRU* combination; e.g., with a 100GB cache size, we observe that the best strategy is 46% better across all the metrics. Finally, we can see that the optimal strategy can be quite different both across the space of metrics and the cache sizes. For instance, *LCD +LRU* is the best strategy w.r.t. hit rate at 1GB cache, but *Cross +LFU* is the best w.r.t. hit rate at 1TB. Similarly, at 1TB, *Cross +LFU*, *LCE +LFU*, and*Prob +LFU* are the best strategies w.r.t. hit rate, traffic reduction, and server load reduction respectively.[3] We do not yet have a simple analytical basis to explain why the optimal combination varies, and we leave this for future work.

Correlation across metrics and cache sizes: Next, we summarize how *correlated* the different strategies are across metrics and cache sizes. Recall that we have 35 different combinations of content placement/replacement algorithms. Now, for a given cache size and metric, we construct the 35-item vector with each entry denoting the performance of a given combination. Then, given these vectors, we compute the Pearson correlation coefficient between these vectors for different cache sizes and metrics. In Figure 8a, we show the correlation between each pair of metrics for a given cache size. Similarly, in Figure 8b, we show the correlation between pairs of cache sizes for each metric.

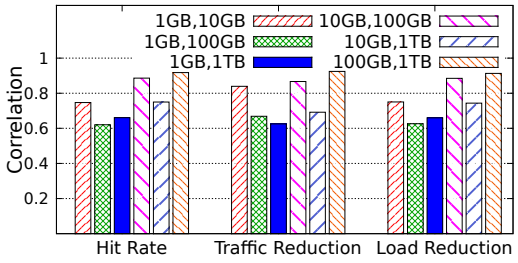
First, we see that hit rate and server load reduction are perfectly correlated. This is expected, as higher cache hit rate reduces server load. Second, we find that optimizing the server load reduction

[3]We found it interesting that Centrality performs poorly across different scenarios. We conjecture that in the strictly hierarchical structure of the Chinese Internet, very few routers (located near the core) have high betweenness and are selected for caching.

does not necessarily translate into a reduction in the overall network footprint. This is also expected, since the traffic reduction depends not only on cache hits, but also on *where* hits occur and whether they occur for popular vs. unpopular objects. We revisit this in greater depth in Section 7, where we systematically dissect the benefits of the various caching strategies. Finally, with respect to the cache sizes, we see a natural correlation between similar cache sizes—1GB cache is the least correlated with all other values, and the correlation between 100GB-1TB caches is the strongest.



(a) Across cache sizes



(b) Across metrics

Figure 8: Hitrate and server load reduction are most correlated, and similar-sized caches have higher correlations.

Is there a dominant strategy? The previous results show that the *best* strategy varies quite significantly across the space of cache sizes and metrics. Given this diversity, a natural question for ICN deployment is whether there are some strategies that are consistently close to the optimal value, even if they are not always the best. To understand this, we calculate an aggregate performance metric that computes the average normalized distance of each strategy from the best possible strategy for each of the 12 scenarios (i.e., 4 cache sizes and 3 metrics). Specifically, for a given scenario s (i.e., a fixed cache size and metric), we compute a score for a given placement-replacement combination pr relative to the performance of the best combination $Perf_s^*$ for that scenario: $Score_{pr,s} = \frac{Perf_s^* - Perf_{pr,s}}{Perf_s^*}$, where $Perf_{pr,s}$ is the performance of pr on scenario s. Then, the overall score $Score_{pr}$ is the mean across all scenarios.

Table 3 shows this normalized score for the 35 caching strategies and highlights the top three strategies. First, we see that in terms of content replacement, *LFU* is the most dominant strategy. In fact, the top 5 strategies all use *LFU*. Second, looking at each dimension separately may not necessarily indicate the best strategy. For example, for cache placement, *Rand* and *LCD* are the top two strategies, but none of their overall combinations rank in the top-3 strategies highlighted in the table. In particular, *LCD* does not work in concert with *LFU*, which, as we saw earlier, is the best strategy for content replacement. Similarly, we find that the combination of the best placement (*Rand*) and the best replacement (*LFU*) strategies

	FIFO	*LFU*	*LRU*	*TTL*	*Size*	RowAvg
LCE	0.64	0.92	0.66	0.76	0.48	0.69
LCD	0.86	0.60	0.88	0.85	0.52	0.74
Rand	0.71	0.91	0.74	0.82	0.62	0.76
Prob	0.65	0.93	0.69	0.78	0.57	0.72
PProb	0.62	0.91	0.67	0.76	0.58	0.71
Centrality	0.38	0.43	0.43	0.40	0.30	0.39
Cross	0.66	0.92	0.70	0.77	0.52	0.71
ColAvg	0.65	0.80	0.68	0.73	0.51	

Table 3: Normalized scores of each strategy across all 12 scenarios to identify a "dominant" strategy that is optimal or close to optimal most of the time.

(i.e., the best row and column averages) is not the optimal strategy; the global best strategy is *Prob +LFU* with a score of 0.93. The *Rand +LFU* combination is also quite close with an averaged normalized score of 0.91.

5.3 Summary of main findings

Our analysis of the impact of video workloads on ICN caching metrics shows that:

- Cache sizes need to be greater than 100GB to provide significant (22% traffic reduction and 34% server load reduction) benefit for PPTV traffic. Compared with other workloads, video seems to require much larger cache sizes; for instance, prior work suggests that a cache that can store 10M files on each router can bring a 20% traffic reduction of BitTorrent content [47]. We also see a natural diminishing returns property wherein increasing the cache 10× from 100GB to 1TB only yields less than 2× improvement.
- The best combination of content placement and replacement depends on the operating regime and the caching metric of interest. For instance, techniques that optimize cache hit rates or server load, do not directly yield a substantial reduction in network footprint. Therefore, ISPs should carefully select their caching strategies according to their objectives (e.g. optimizing traffic reduction or server load reduction) and their network characteristics (e.g. cache size, hierarchical/flat topology).
- Fortunately, there exists a sufficiently dominant strategy *Prob +LFU* that is close to the best possible strategy almost always. We also find that *LFU* is the dominant replacement strategy. However, the best content placement strategy is not part of the best combined placement-replacement strategy.

6. VIDEO QUALITY OF EXPERIENCE WITH ICN

As discussed in prior work, Internet video introduces new QoE considerations such as buffering ratio, startup delay, and average delivered bitrate [21, 34]. In this section, we study the effect of using ICN on video QoE using the same simulation setup as described in Section 5.1. Specifically, we are interested in understanding:

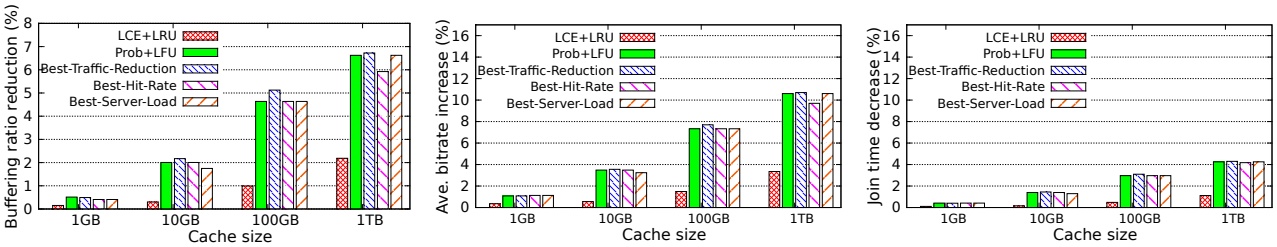
- How do the ICN caching schemes and cache sizes impact video QoE?
- How do the network-level metrics and the best choices from the previous section translate into video QoE considerations?

6.1 Methodology

To understand QoE improvements of ICN, first, we need a way of modeling the QoE that a client would perceive if its requests were served from a specific in-network cache. In the absence of a real caching deployment, we do not have this directly; thus, we need some way to extrapolate the QoE metrics from our PPTV traces.

Province	Buffering ratio			Avg. bitrate			Join Time		
	SamePoP	SameISP	DiffISP	SamePoP	SameISP	DiffISP	SamePoP	SameISP	DiffISP
BJ	1	1.17	2.69	1	0.85	0.69	1	1.07	1.48
LN	1	1.33	4.01	1	0.92	0.56	1	1.03	1.49
ZJ	1	1.18	3.35	1	0.80	0.47	1	1.05	1.71
...	...	...	...	...	...	...	...	...	...
Others	1	1.17	3.18	1	0.86	0.55	1	1.06	1.52

Table 4: Normalized QoE metrics across different provinces for different server classes.



(a) Buffering ratio (b) Average bitrate (c) Join time

Figure 9: Improvement in different QoE metrics for different ICN caching strategies and cache sizes.

Given a specific client-router pair c, r, our extrapolation methodology tries to find requests in the original PPTV traces that share *similar* client and server attributes [41]. For instance, if there is a client c from Beijing on a DSL connection from China Telecom as its ISP, and the router r chosen in the ICN caching case is also a China Telecom router in Beijing, we select all PPTV traces where the client-server attributes match these values and use those statistics to extrapolate the QoE for the hypothetical c, r request. Then, to predict the expected QoE for this hypothetical request c, r, we use the median value over requests with similar attributes. (For all metrics other than buffering ratio, we use the median. For buffering ratio, the median is typically zero, and we use the average.)

Ideally, we would like to pick the requests that match every possible client and server attributes. However, we run into a classic "curse of dimensionality" problem where the data becomes quite sparse as we increase the number of dimensions. Furthermore, for some provinces (e.g., TB for Tibet), client terminals (e.g,. mobile), and client connection types (e.g., 3G), we have only few requests.

Given the above constraints, we take the following heuristic. First, we only focus on the dominant client terminal and connection types; in this case it is desktop clients on broadband connections. Second, we exploit the hierarchical structure of the Chinese Internet topology to focus on three *classes of servers* rather than attempt to match all possible server attributes. (The three main ISPs, ChinaTelecom, ChinaUnicom and China Mobile organize their networks in a hierarchical manner and connect to one another at a higher level. Other small ISPs interconnect via one of these major ISPs.) Specifically, we partition the content-view dataset into three non-overlapping categories based on the locations of the client and the server IPs:

1. *Same PoP (SamePoP):* The client and the server are in the same PoP (point of presence) of the same ISP;
2. *Same ISP, but different PoPs (SameISP):* The client and the server are in the same ISP network but in different PoPs;
3. *Different ISPs (DiffISP):* The client and the server are in different ISPs.

Finally, we use the client location at a province-level granularity. To this end, for each province with a sufficiently large number of requests, we consider the servers in one of the above three categories and obtain the summary statistics. For the remaining provinces that do not have individually a sufficient number of client requests in our dataset, we use the average value across the provinces with sufficient data for each of the three server categories.

Using this extrapolation methodology, Table 4 shows the improvement of the QoE metrics relative to the case of SamePoP for 3 of the major provinces. Due to confidentiality considerations, we cannot show the absolute QoE numbers and normalize the values relative to the baseline of SamePoP; e.g., for SameISP the normalized QoE will be $NormQoE_{SameISP} = \frac{QoE_{SameISP}}{QoE_{SamePoP}}$. Note that for buffering ratio and join time, higher values of the normalized metric indicate worse QoE and for average bitrate, lower values indicate worse QoE. We observe that the QoE is strongly dependent on the distance between the user and the provider. For instance, in the "Others" row the buffering ratio increases by 17% if the server is in a different PoP but within the same ISP; this value grows by 218% for the case of different ISPs.

6.2 Results

Given the above extrapolation methodology, we can compare the QoE provided by a specific ICN strategy relative to the QoE provided by the reference, no-cache strategy. That is, for each request q, we calculate two normalized QoE values based on the values shown in Table 4: $NormQoE_{q,ICN}$ and $NormQoE_{q,NoCache}$. Then, we calculate the difference between the per-request means of these values: $\frac{\sum_{q=1}^{V} NormQoE_{q,NoCache}}{V} - \frac{\sum_{q=1}^{V} NormQoE_{q,ICN}}{V}$, where V is the total number of video requests.[4]

Figure 9 shows this improvement metric (expressed as a percentage). In the interest of brevity, we only consider a few interesting caching strategy combinations from the previous section. Specifically, we consider three combinations: (1) *LCE +LRU*, which is the most basic combination suggested in previous work; (2) *Prob +LFU*, which emerged as the dominant strategy across all scenarios from the previous section; and (3) The best caching strategy for each of the different network-level metrics from the previous section.

We make three main observations from this figure. First, the overall magnitude of QoE improvement is quite low; the maximum performance gap between the largest 1TB cache and a no-ICN scenario is $\leq$ 12% across all QoE metrics. Second, with small caches $\leq$ 10GB, video QoE improvement is almost negligible. Finally, we see that the more complex caching strategies are more beneficial

[4]For bitrate, the difference is reversed since a higher value is better.

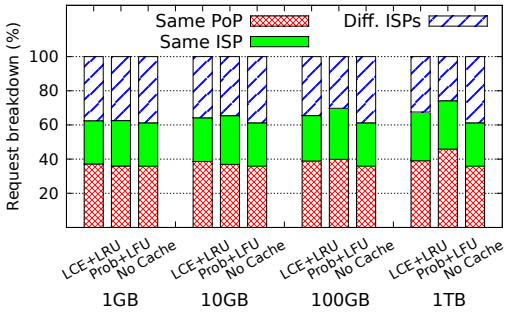


Figure 10: Breakdown of requests served by servers in three different types of locations.

to QoE than simpler strategies. For example, the average relative performance improvement by using the *Prob +LFU* strategy compared to *LCE +LRU* strategy across all the metrics and cache sizes is close to 4×.

Overall, we find that the traffic reduction appears to be the most useful criterion to improve the QoE metrics. This is intuitively expected, as reducing traffic load improves requests locality. As we saw earlier in Table 4, locality directly correlates with improved QoE in our dataset. We also see that the dominant strategy from the previous section (i.e., *Prob +LFU*) is again very competitive, as it is close to the best result in all cases.

To further analyze why the overall improvement in the QoE seems quite low, we provide a breakdown of the number of requests served by the three key categories of (client relative) provider locations in Figure 10. As a point of reference, we also show the corresponding breakdown for the case of no ICN deployment (denoted by No Cache in the figure). For ease of visualization, we focus only on two of the ICN strategies. The figure shows that the current server deployment of PPTV is effectively *localizing* requests; i.e., the fraction of SamePoP and SameISP requests is largely similar at $\leq$ 100GB caches, and only at 1TB does *Prob +LFU* show a visible difference.

6.3 Summary of key findings

After analyzing the effect of ICN caching strategies on video QoE improvement, we find that:

- We see non-trivial QoE improvements only with caches $\geq$ 100GB. Even then, the QoE improvement is quite low $\leq$ 12% across all QoE metrics even with a 1TB cache.
- The caching strategy that optimizes the traffic reduction appears to be the best strategy to also optimize QoE; the dominant strategy in terms of the caching metrics (i.e., *Prob +LFU*) is again a close-to-optimal strategy with respect to QoE (as shown in Figure 9).
- The QoE improvement seems to be low mainly because of the good coverage of the PPTV servers across popular PoPs/ISPs. This suggests that for popular video services whose server footprint is already spatially diverse, the QoE benefits of ICN will be limited.

7. DISSECTING BENEFITS

In the previous two sections, we studied the aggregate improvement of different content replacement/placement algorithms in terms of network-level and QoE metrics. In this section, we provide a more in-depth analysis to identify *how* and *where* these benefits manifest, so that it can better inform the deployment decisions of content providers and ISPs as they consider ICN-like architectures. Specifically, we are interested in:

- Which network routers (e.g., edge vs. core) provide the largest contribution to caching metrics and QoE improvements?
- What types of requests (e.g., low vs. high popularity) and client locations (e.g., types of provinces) are more likely to perceive the largest benefits?
- At what times of day do we see the most benefits?

In the interest of brevity, we only focus on *Prob +LFU* that we found to be the dominant strategy across caching metrics and near-optimal for QoE.

7.1 Routers contributing to improvement

We identify the (network-wide) top-100 routers with respect to performance metrics (i.e., hitrate, contribution to traffic reduction, and contribution to server load reduction). Then, we characterize these top-100 routers along two dimensions: (1) The relative role of a router in the network (i.e., Edge, Middle, or Core) and (2) Geographical location of the router (i.e., which province). Because we do not have such role annotations in our measured topology, we use the following heuristic for (1). First, we classify a router's role for each request relative to the shortest path between the client and the server. We divide the path into three equal-sized segments: Edge, Middle, or Core, with Edge being the segment closest to the client, Core being close to the server, and Middle being the rest. Then, for each router, we compute its overall role by taking a simple majority vote over all requests the router is involved. For (2), we use a proprietary IP-to-location mapping database [2].

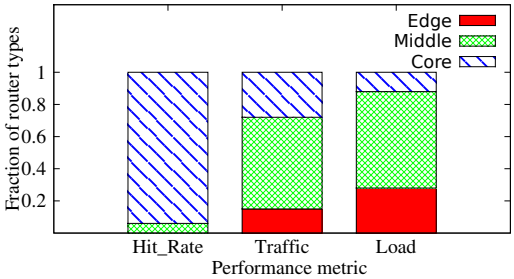
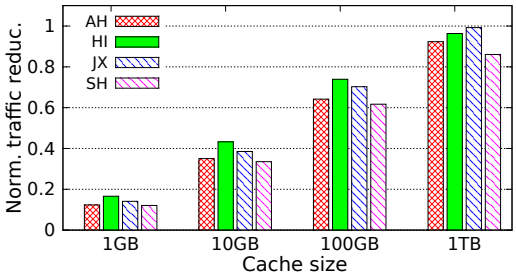


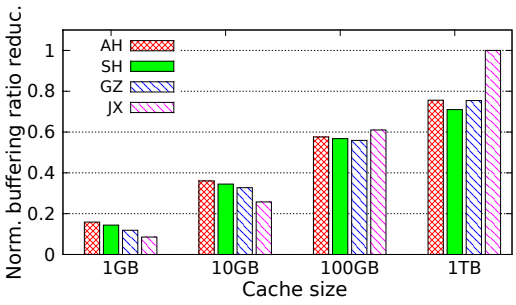
Figure 11: Breakdown of top routers locations.

Figure 11 shows the breakdown of these routers in terms of their network roles. We see that most of the top-100 routers with high cache hit rate are Core routers. This is because the requests of the same content aggregate when going towards the direction of the origin server. However, Core routers contribute very little to reducing the traffic and server load. This observation is consistent with previous work on more general request workloads [23]. We posit that the incentives for real world ICN deployment by ISPs and content providers will likely stem from traffic and load reduction. Thus, we suspect that ICN routers will not be prominent in the network core.

We have also analyzed how these top-100 routers are geographically spread (not shown). We observed two main trends. First, the top routers with respect to traffic and server load reductions are mainly in provinces with a large client population (e.g., HE, SD, GD). This is expected, as the routers in regions with larger populations serve more client requests and thus cumulatively reduce more network footprint. Second, top routers with respect to hit rate improvement are located in provinces with many PPTV servers (e.g., JL, BJ, SC). This is consistent with Figure 11, as a province with more servers naturally has more routers closer to servers. Similar to our earlier observation in Section 6 that the video QoE metrics are



(a) Traffic reduction



(b) Buffering improvement

Figure 12: Locations with best improvement in traffic reduction and buffering ratio improvement.

closely correlated with traffic reduction, here, we saw the routers contributing to most QoE improvements are, to a great extent, the same routers that contribute most to traffic reduction (not shown).

7.2 Request-level analysis

Locations with most improvement: We identify the regions with the highest caching performance and QoE improvements using the traffic reduction and buffering ratio as representative caching- and QoE-related metrics. Recall that each request q is associated with the province where the client is located. For each province, we calculate the average traffic reduction and QoE improvement across requests originating in that province. Then, we normalize these values across different provinces with respect to the province with the highest value such that the best value is 1. While this normalization is because of PPTV's confidentiality constraints, since our goal is to study the relative behaviors across provinces, we believe this is a reasonable choice.

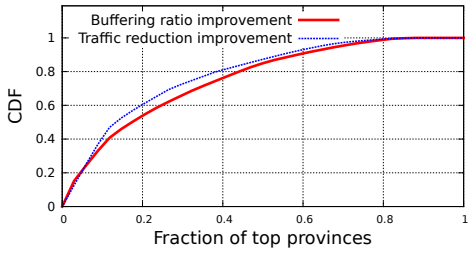
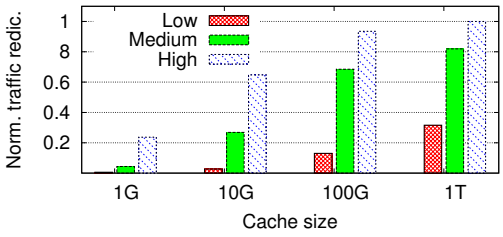
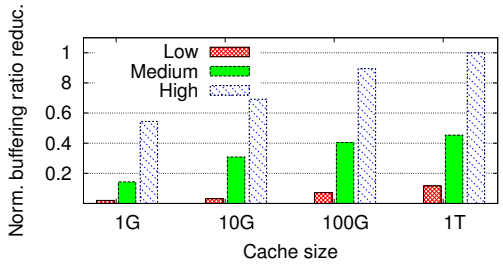


Figure 13: CDF of buffering ratio and traffic improvement as functions of top provinces.

Figure 12 shows the normalized values for top 4 provinces. The top provinces shown in Figure 12a and Figure 12b have substantial overlap, with AH, JX and SH being in common. For QoE improvement, the top provinces tend to be those with few servers deployed (e.g., JX, AH, GZ, SH) (see Figure 2). This is intuitively



(a) Traffic reduction



(b) Buffering improvement

Figure 14: Improvement in video QoE and traffic reduction vs. popularity of objects.

expected from the earlier observation in Figure 10: Provinces with PPTV servers already localize traffic and cannot substantially improve QoE further.

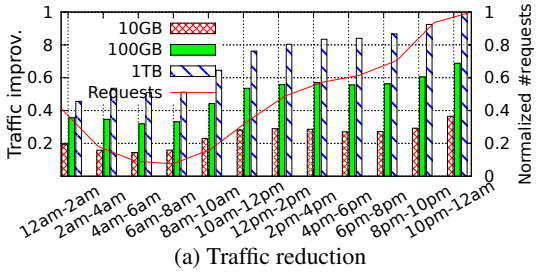
The previous result looks at the average per-request improvement, but does not consider the volume of requests from a given region. To account for this, we also analyzed how the top-k provinces contribute to the *total* improvement across requests $\sum_q Improvement_q$ for different $Improvement$ metrics. Figure 13 shows the cumulative distribution of improvement of buffering ratio and traffic reduction as a function of top provinces. We see that there is a skewed and diminishing returns behavior: the top-20% provinces contribute about 60% of the total improvement. As discussed earlier, these tend to be provinces with high request volumes but low/no server presence. The above results suggest that ISP deployment efforts may be better invested in such regions to gain maximum return-on-investments.

Request popularity: In Figure 14, we analyze which type of requests receive the highest QoE improvement and contribute most to traffic reduction. To this end, we categorize videos into three popularity bins: high ($\geq$ 10,000 views per day), medium (5,000 to 10,000 views per day), and low ($<$ 5,000 views per day). Then we assign each request q to one of these three bins and, for each bin, we compute the average values of traffic and buffering ratio improvements. Then we normalize these values such that the high-popularity bin using 1TB cache is assigned a value of 1. The result is quite intuitive: popular contents by virtue of higher request rates and cache hits see the highest QoE improvements and contribute most of the traffic reduction. With increasing the cache size, we observe that objects with medium and low popularity also benefit.

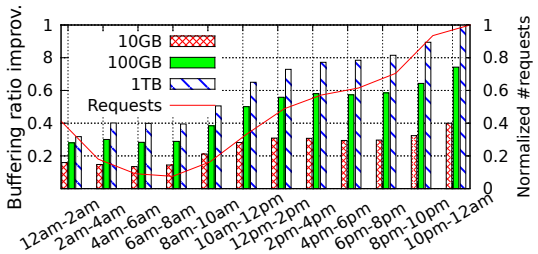
We have also analyzed the contribution of the top-k popular objects to the aggregate improvement of buffering ratio and traffic reduction. Here, we observed an even greater skew; the top 0.0014% objects (≈700 videos) contribute 91% of the buffering ratio improvement and 92% of the traffic improvement (not shown).

7.3 Temporal analysis

Finally, we analyze at *what time of day* we observe most of the perceived benefit. (Since there is only one time zone in China, we do not need to be concerned about the effect of different time



(a) Traffic reduction



(b) Buffering

Figure 15: Time-of-day effects: We can visually confirm that the times of high improvement roughly coincides with high request volume.

zones.) Figure 15 shows the breakdown of the QoE improvement for the buffering ratio and traffic reduction per content-view trace as functions of time on a single day. For brevity, we do not show the other days or QoE metrics since the results are similar. We observe that, on average, most QoE improvement and traffic reduction is achieved in the evening (8:00pm to 12:00am). We can see that these periods are actually the busiest time (i.e., highest number of requests) for PPTV platform. This observation that caching provides the most gain at the time interval with most video traffic provides good incentives for access ISPs, as reducing peak utilization can reduce costs with "burstable billing" schemes that charge based on 95-th percentile usage.

8. DISCUSSION

Cache placement and provisioning: Our evaluation assumes that all locations have the same amount of cache storage. A natural question is the impact of more fine-grained cache placement and provisioning strategies on our findings. For example, previous work has shown that a network-wide cache deployment may not yield substantial benefits [23]. While an exploration of optimal cache placement is outside the scope of this paper, we present some initial results.

We make a performance comparison between two extreme cache provisioning strategies: (1) *Ubiquitous caching*, where every router in the network is equipped with a cache of the same size, and (2) *Access-Only caching*, where only access routers (i.e., one hop away from the client) are equipped with caches. To make a fair comparison between these two strategies, we assume that the total volume of network-wide cache storage is the same.

Our experiments show that the relative gain of ubiquitous caching over access-only caching depends on the cache size. For a very small cache size of 1GB, the gain is as large as $2\times$ for different performance metrics. As the cache size increases, however, the performance gain diminishes. This is consistent with previous results reported in [23].

Origin server placement: As we saw earlier in Section 6, the improvements of video QoE as yielded by ICN caching are quite marginal, at least partially, because of the sufficient geodiversity of PPTV servers. A natural question, therefore, is the sensitivity of our results to the number and placement of origin servers.

To find the answer, we evaluate the sensitivity of our results to the number of origin PPTV servers. The results are shown in Figure 16. In each plot, the X axis represents the total number of origin servers and the Y axis is the gain yielded by ICN caching for the corresponding QoE metric. We show the normalized values on the Y axes to make it easier to compare the results across different cache sizes and metrics. The locations of each given total number of servers are chosen at random from all the 221 PPTV servers.

We observe that as the number of origin servers grows, the effect of ICN caching on improving QoE metrics diminishes because of increasing geodiversity of the content servers. We believe that our results in Figure 9 are representative—Popular content providers today (e.g., YouTube, Netflix) that carry the most amount of traffic, typically, already have a significant geographic footprint [16], if not directly, at least via CDNs.

Request routing: One question is whether intelligent request routing strategies can help; e.g., would some of the less optimal caching strategies we saw perform better under different routing regimes? Extrapolating from prior work, we posit that complex request routing will offer minimal benefits over simple routing [23,46] and will likely not impact our observations.

Other forms of video: We have studied the interaction between VOD workloads and ICN because it is the dominant form of Internet video used today. An interesting direction of future work is to replicate our analysis on other types of video workloads (e.g., live streams).

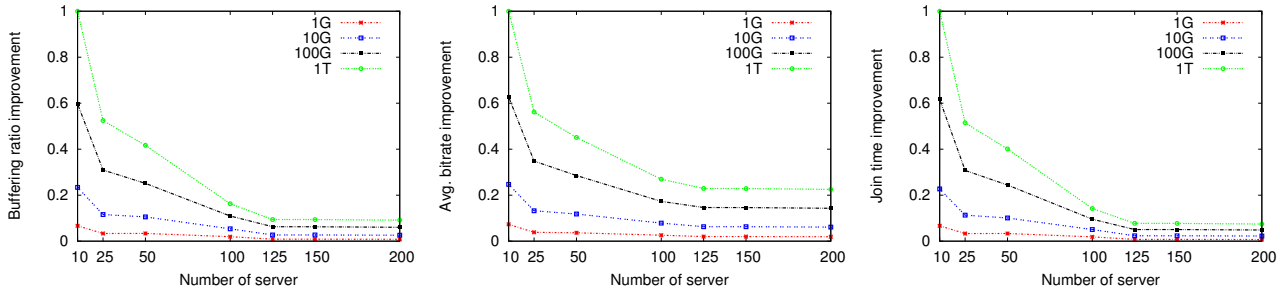
9. CONCLUSIONS

Even though video and ICN have been considered natural allies, there has been little work on analyzing how video workloads impact the network-level benefits of ICN and how ICN helps improve video QoE metrics. Our work addresses this crucial gap. To the best of our knowledge, our study is the most comprehensive and the largest-scale analysis of ICN-on-video in terms of trace size, topology scale, and the space of ICN caching strategies explored.

Our findings have key implications for providers and future Internet architectures: (1) the benefits of ICN for video workloads with caches smaller than 100GB are marginal; (2) the combined content placement and content replacement strategy of *Prob +LFU* is close to the optimal with respect to improvements in both ICN caching performance and QoE metrics across the entire spectrum of analysis parameters; and (3) a small number of regions without origin servers and with requests for highly popular objects contribute to the most of the improvements. That said, this is by no means the final word in this space, and there are several directions for future work including analyzing other forms of video (e.g., live stream and short VOD), providing a more analytical basis for our empirical observations, and exploring the effects of different ICN routing schemes.

Acknowledgments

The authors would like to thank Alexander Afsanyev for helping us gain an in-depth understanding of ndnSIM. We also thank Stratis Ioannidis for shepherding our paper and Haiyong Xie for insightful discussions. This work is supported in part by the National Basic Research Program (2012CB315802) and the Natural Science Foundation of China (61379133). This work is also funded in part by NSF under award number CNS-1345305.



(a) Buffering ratio (b) Average bitrate (c) Join time

Figure 16: Improvement of QoE by ICN as a function of number of origin servers.

10. REFERENCES

[1] CCNx Project. http://www.ccnx.org/.

[2] Chunzhen. http://www.cz88.net/.

[3] Cisco forecast. http://blogs.cisco.com/sp/comments/cisco_visual_networking_index_forecast_annual_update/.

[4] COntent Mediator architecture for content-aware nETworks (COMET). http://www.comet-project.org/.

[5] Direct Code Execution (DCE). http://www.nsnam.org/docs/dce/release/1.0/manual/singlehtml/index.html.

[6] NetInf. http://www.netinf.org/.

[7] PPCloud: Big Data in PPTV. http://www.dvbcn.com/2014/03/18-109253.html.

[8] PPTV. http://www.pptv.com/.

[9] M. Abrams, C. R. Standridge, G. Abdulla, E. A. Fox, and S. Williams. Removal policies in network caches for world-wide web documents. In *Proc. SIGCOMM*, 1996.

[10] A. Afanasyev, I. Moiseenko, and L. Zhang. ndnSIM: NDN simulator for NS-3. http://named-data.net/techreport/TR005-ndnsim.pdf, 2012.

[11] S. Ando and A. Nakao. In-network cache simulations based on a youtube traffic analysis at the edge network. In *Proc. CFI*, 2014.

[12] P. A. Aranda, M. Zitterbart, Z. Boudjemil, M. Ghader, G. H. Garcia, M. Johnsson, A. Karouia, G. Lazar, M. Majanen, P. Mannersalo, D. Martin, M. T. Nguyen, S. P. Sanchez, P. Phelan, M. P. de Leon, G. M. Sollner, Y. Zaki, and L. Zhao. 4WARD. http://www.4ward-project.eu/, 2010.

[13] B. Cheng, X. Liu, Z. Zhang, and H. Jin. A measurement study of a peer-to-peer video-on-demand system. In *Proc. IPTPS*, 2007.

[14] A. Balachandran, V. Sekar, A. Akella, and S. Seshan. Analyzing the potential benefits of cdn augmentation strategies for internet video workloads. In *Proc. IMC*, 2013.

[15] C. M. Cabral, C. E. Rothenberg, and M. F. Magalhães. Mini-CCNx: Fast prototyping for named data networking. In *Proc. ICN*, 2013.

[16] M. Calder, X. Fan, Z. Hu, E. Katz-Bassett, J. Heidemann, and R. Govindan. Mapping the Expansion of Google's Serving Infrastructure. In *Proc. IMC*, 2013.

[17] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system. In *Proc. IMC*, 2007.

[18] W. K. Chai, D. He, I. Psaras, and G. Pavlou. Cache "less for more" in information-centric networks. In *Proc. IFIP Networking*, 2012.

[19] H. Che, Y. Tung, and Z. Wang. Hierarchical web caching systems: modeling, design and experimental results. *IEEE JSAC*, 2002.

[20] R. Chiocchetti, D. Rossi, and G. Rossini. ccnsim: An highly scalable ccn simulator. In *Proc. ICC*, June 2013.

[21] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang. Understanding the impact of video quality on user engagement. In *Proc. SIGCOMM*, 2011.

[22] S. Eum, K. Nakauchi, M. Murata, Y. Shoji, and N. Nishinaga. Catt: Potential based routing with content caching for icn. In *Proc. ICN*, 2012.

[23] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, and S. Shenker. Less pain, most of the gain: Incrementally deployable ICN. In *Proc. SIGCOMM*, 2013.

[24] C. Fricker, P. Robert, J. Roberts, and N. Sbihi. Impact of traffic mix on caching performance in a content-centric network. In *Proc. INFOCOM WKSHPS*, 2012.

[25] M. Gritter and D. R. Cheriton. An architecture for content routing support in the Internet. In *USITS*, 2001.

[26] B. Han, X. Wang, N. Choi, T. Kwon, and Y. Choi. Amvs-ndn: Adaptive mobile video streaming and sharing in wireless named data networking. In *Proc. NOMEN*, 2013.

[27] D. Han, A. Anand, F. Dogar, B. Li, H. Lim, M. Machado, A. Mukundan, W. Wu, A. Akella, D. G. Andersen, J. W. Byers, S. Seshan, and P. Steenkiste. Xia: Efficient support for evolvable internetworking. In *Proc. NSDI*, 2012.

[28] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. In *Proc. SIGCOMM*, 2014.

[29] Y. Huang, T. Z. Fu, D.-M. Chiu, J. C. Lui, and C. Huang. Challenges, design and analysis of a large-scale p2p-vod system. In *Proc. SIGCOMM*, 2008.

[30] V. Jacobson et al. Networking named content. In *CoNext*, 2009.

[31] V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, K. claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh. Named Data Networking (NDN) project. http://named-data.net/techreport/TR001ndn-proj.pdf, 2010.

[32] J. Jiang, V. Sekar, and H. Zhang. Improving fairness, efficiency, and stability in http-based adaptive video streaming with FESTIVE. In *Proc. CoNEXT*, 2012.

[33] T. Koponen et al. A data-oriented (and beyond) network architecture. In *SIGCOMM*, 2007.

[34] S. S. Krishnan and R. K. Sitaraman. Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs. In *Proc. IMC*, 2012.

[35] D. Kulinski, J. Burke, and L. Zhang. Video Streaming over Named Data Networking. *IEEE COMSOC MMTC E-Letter*, 2013.

[36] N. Laoutaris, H. Che, and I. Stavrakakis. The LCD interconnection of LRU caches and its analysis. *Perform. Eval.*, 2006.

[37] N. Laoutaris, S. Syntila, and I. Stavrakakis. Meta algorithms for hierarchical web caches. In *Proc. ICPCC*, 2004.

[38] Z. Li, M. Sbai, Y. Hadjadj-Aoul, A. Gravey, D. Alliez, J. Garnier, G. Madec, G. Simon, and K. Singh. Network friendly video distribution. In *Proc. NOF*, 2012.

[39] Z. Li and G. Simon. Time-shifted tv in content centric networks: The case for cooperative in-network caching. In *Proc. ICC*, 2011.

[40] H. H. Liu, Y. Wang, Y. R. Yang, H. Wang, and C. Tian. Optimizing cost and performance for content multihoming. In *Proc. SIGCOMM*, 2012.

[41] X. Liu, F. Dobrian, H. Milner, J. Jiang, V. Sekar, I. Stoica, and H. Zhang. A case for a coordinated internet video control plane. In *Proc. SIGCOMM*, 2012.

[42] Z. Ming, M. Xu, and D. Wang. Age-based cooperative caching in information-centric networks. In *Proc. INFOCOM WORKSHOPS*, 2012.

[43] I. Psaras, W. K. Chai, and G. Pavlou. Probabilistic in-network caching for information-centric networks. In *Proc. ICN*, 2012.

[44] I. Psaras, R. G. Clegg, R. Landa, W. K. Chai, and G. Pavlou. Modelling and evaluation of ccn-caching trees. In *Proc. NETWORKING*, 2011.

[45] G. Rossini and D. Rossi. A dive into the caching performance of content centric networking. In *Proc. CAMAD*, 2012.

[46] A. Sharma, A. Venkataramani, and R. K. Sitaraman. Distributing content simplifies isp traffic engineering. In *Proc. SIGMETRICS*, 2013.

[47] G. Tyson, S. Kaune, S. Miles, Y. El-khatib, A. Mauthe, and A. Taweel. A trace-driven analysis of caching in content-centric networks. In *Proc. ICCCN*, 2012.

[48] W. Wang, S. Yi, Y. Guo, M. A. Kaafar, and Z. Li. Crcache: Exploiting the correlation between content popularity and network topology for icn caching. In *Proc. ICC*, 2014.

[49] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie. Optimal cache allocation for content-centric networking. In *Proc. ICNP*, 2013.

[50] H. Yin, X. Liu, F. Qiu, N. Xia, C. Lin, H. Zhang, V. Sekar, and G. Min. Inside the bird's nest: Measurements of large-scale live vod from the 2008 olympics. In *Proc. IMC*, 2008.

[51] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng. Understanding user behavior in large-scale video-on-demand systems. In *Proc. EuroSys*, 2006.