PAINT: Partial In-Network Transcoding for Adaptive Streaming in Information Centric Network

Yichao Jin and Yonggang Wen
School of Computer Engineering
Nanyang Technological University
{yjin3, ygwen}@ntu.edu.sg

***Abstract*—Information centric network (ICN) has emerged as a promising architecture to efficiently distribute content over the future Internet. However, ICN proposals may still not be cost efficient enough for adaptive video streaming. The problem is, each ICN node caches duplicated copies of the same content for each bitrate version in its limited storage space. Thus the cache hit ratio drops, and the bandwidth cost of serving the cache missed requests increases. This paper proposes PAINT (Partial In-Network Transcoding) scheme to reduce the operational cost of delivering adaptive video streaming over ICN. Specifically, we consider both the in-network caching and transcoding services at each ICN node, where the storage and transcoding resources can be dynamically scheduled. Then we formulate an optimization problem to balance the trade-off between the transcoding and bandwidth costs. Next we analytically derive the optimal strategy, and quantify cost savings compared with existing schemes. Finally, we verify our solution by intensive numerical evaluations. The results indicate PAINT can achieve significant cost savings (e.g., up to 50% in typical scenarios). Besides, we find the optimal strategy and the cost savings can be affected by the cache capacity, the unit price ratio, the hop distance to origin server, and the Zipf parameter of users' request patterns.**

I. Introduction

Nowadays, a remarkable amount of adaptive video streaming data is dominating the Internet traffic, largely driven by the ubiquitous integration between various devices and user-friendly streaming services [1]. In particular, real-time video/audio streaming traffic via fixed access network was around 25,000 Petabytes, accounting for 58.6% of the total Internet usage, in the US in 2013. This number is expected to reach over 150,000 Petabytes, which will be 66.0% of the total Internet traffic in 2018 [1]. Indeed, those streaming data are adaptively delivered in different formats and bitrates to different devices, including PCs (27%), smart TVs (30%), smartphones and tablets (29%) [2].

Information Centric Networks (ICN) [3], [4] has emerged as a promising framework to efficiently distribute video streams over the future Internet [5]. Specifically, ICN adopts name-based routing to enable a novel "host-to-content" model, where every content is uniquely identified and accessed without being associated to a host address. This paradigm intrinsically supports in-network caching, shifting the content from the origin server to a closer location to users. As a result, it offers an opportunity to significantly reduce the operational cost of delivering adaptive video streaming.

978-1-4799-4852-9/14/$31.00

However, current ICN proposals are not cost efficient to distribute adaptive video streaming to heterogeneous end devices. The main problem is that, under current ICN framework, different versions (e.g., formats, resolutions, and bitrates) of the same segment are with different names [6]. Consequently, they are cached as different content. This reduces the cache hit ratio of different video segments at each ICN node, given the limited in-network caching space. It eventually increases the bandwidth cost of serving those cache misses. Note that, this paper defines the cache hit ratio in an ICN node as the percentage of requests that are served either directly from the cache or based on local in-network transcoding.

In this paper, we propose PAINT (PArtial In-Network Transcoding) as a novel scheme to reduce the operational cost of delivering adaptive video streaming services over ICN. Specifically, PAINT integrates real-time streaming transcoders [7], [8], which are commonplace nowadays [9], into ICN routers. In this way, ICN nodes only need to cache the highest bitrate version for a partial set of segments, and derive all other representations for them based on local online transcoding. This improves the cache hit ratio under the constrained local cache space, saving the bandwidth cost of delivering multiple versions of a same segment. Nevertheless, such transcoding services may generate additional transcoding cost, which would overwhelm the savings if not appropriately scheduled. Therefore, we need to intelligently allocate the storage and transcoding resources at each ICN node, with an objective to minimize the total operational cost.

Our contributions of this paper are multifolder, including:

- We propose PAINT with an objective to minimize the total cost by balancing the trade-off between the transcoding and bandwidth costs.
- We formulate the cost-minimization problem as a constrained convex optimization problem, and derive a closed form solution of the optimal strategy on provisioning in-network caching and transcoding resources.
- Through intensive numerical evaluations, we observe the derived strategy can achieve significant cost savings (e.g., up to 50% in typical scenarios) compared with existing schemes in ICN. In addition, we also find both the optimal strategy and cost savings can be affected by various system parameters including, the cache capacity, the unit price ratio, the hop distance to origin server, and the Zipf parameter of users' request patterns.

These insights would offer operational guidelines to design a more efficient ICN architecture, which is able to serve the adaptive video streaming services and those related applications (e.g., cloud social TV [10]–[12]) at a lower cost. This potentially eases the adoption of ICN for the future Internet.

The rest of this paper is organized as follows. Section II outlines the related works. Section III discusses the system architecture and proposes PAINT. Section IV presents problem formulation. Section V theoretically analyzes the optimal strategy and cost savings. Section VI presents numerically evaluations. Finally, Section VII summarizes this work.

II. Related Works

In-network caching, as one of the key components in ICN, had attracted significant attention. Jacobson *et al.* [3] were the first one to systematically propose Content Centric Network (CCN) as a novel and efficient network architecture. They proposed the classic ubiquitous in-network caching scheme, which was also widely adopted in ICN. More recently, the performance of such classic scheme was questioned, and a number of works attempted to improve it. Specifically, Fayazbakhsh *et al.* [13] indicated that, the pervasive caching scheme is not fundamentally necessary for an efficient ICN based on trace-driven study. Chai *et al.* [14] proposed to cache the content only in a subset of nodes. Their results showed the overall storage cost is dramatically saved, while the performance remains at the same level. Li *et al.* [15] described another strategy of coordinated caching scheme to improve the cost efficiency by balancing the trade-off between routing performance and the coordination cost. These works inspired us to rethink the in-network caching scheme in ICN.

Some pioneer works started to focus on distributing adaptive video streaming over ICN. Lederer *et al.* [5] presented a survey on those existing and potential solutions, and claimed such technologies would be practical and useful. Indeed, from the engineering aspect, Kulinski *et al.* [6] had successfully implemented NDNVideo as a prototype to distribute streaming video over ICN. From the research innovation aspect, Grandl *et al.* [9] suggested a pure in-network transcoding scheme to store only the highest quality segment and derive others by real-time transcoding, aiming to overcome the excessive cache usage in ICN nodes. These works motivated us to further investigate this topic for ICN design.

Our work clearly differs from the above related research. In particular, we focus on the cost optimal partial transcoding scheme to efficiently deliver adaptive video streaming, where the problem is how to intelligently schedule in-network caching and transcoding resources at each ICN node. In addition, we develop an analytic framework to balance the trade-off between the transcoding and bandwidth costs, and derive the optimal strategy. Moreover, we study the impact of different system parameters through intensive numerical evaluations. To the best of our knowledge, this work is the first attempt to systematically introduce the partial in-network transcoding scheme to information centric networks.

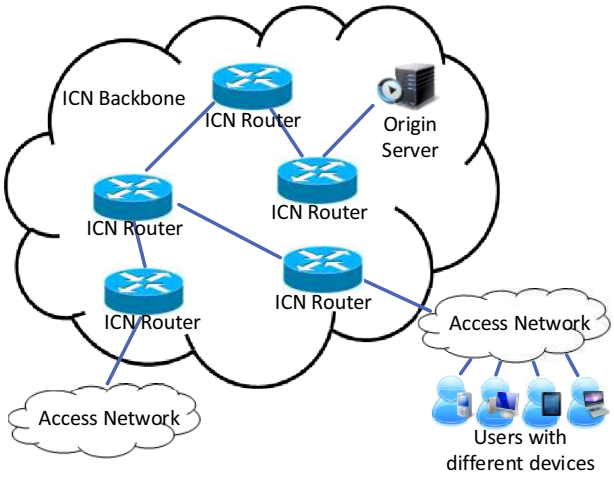


Fig. 1: Adaptive live streaming over ICN architecture

III. System Overview & Proposed Approach

In this section, we first describe the system overview to provide necessary background. Then we propose the system framework and workflow of PAINT. Finally, we present the revised ICN routing scheme after applying PAINT.

A. System Architecture

Figure 1 shows a systematic end-to-end view of delivering adaptive video streaming over ICN architecture. It mainly consists of three parts, including the ICN backbone network, the origin server, and a number of users with different devices. Specifically, all the video segments are originally published by an origin server. They are delivered to the users over an ICN backbone network. Each ICN node is a backbone service entry point attached with an access network. They run the ICN name-based publish/subscribe protocol [3], and support in-network caching. We assume a homogeneous cache size model [16], where each node has the same cache capability C. Moreover, they can be configured to online transcode a subset of the cached video segments. Note that, an ICN node cannot cache all the segments, because its local cache space is usually much smaller than the total volume of content.

One of the critical design objectives is to minimize the total operational cost incurred by delivering adaptive video streams over ICN backbone. Indeed, the operational cost highly depends on the in-network transcoding configuration on each router. In particular, on the one hand, transcoding increases the cache hit ratio and reduces the bandwidth consumption in the network by making more content available in the cache. On the other hand, it increases the workload on the cache and creates a transcoding cost. As a result, there is an opportunity to reduce the total operational cost by examining the trade-off between the transcoding and bandwidth costs.

B. PAINT (Partial In-network Transcoding)

We consider two existing in-network caching and transcoding strategies and propose PAINT as a hybrid solution. First,

TABLE I: Processing strategy comparison

Strategy	Cache Hit Ratio	Transcoding Cost	Bandwidth Cost
All Rate Caching [3]	Lowest	Lowest	Highest
Pure Transcoding [9]	Highest	Highest	Lowest
PAINT	In-between	In-between	In-between

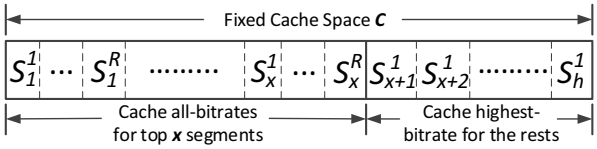
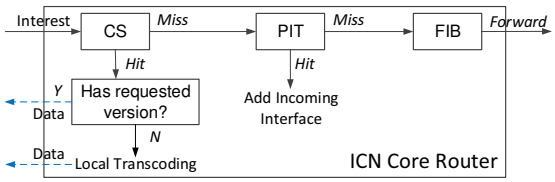
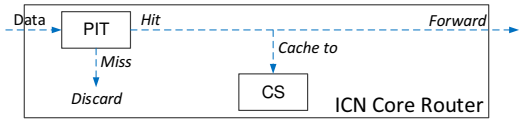


Fig. 2: Illustration of how an ICN node caches video segments under partial in-network transcoding scheme



(a) Handling interest packet (control plane)



(b) Handling data packet (data plane)

Fig. 3: Packet processing engine in ICN router

we look at the classic "*all rate caching*" scheme as in [3]. Under this framework, different bitrate versions of a same video segment are treated as different segments. Thus, all of them are cached by all the traversed ICN nodes. Second, we consider the "*pure transcoding*" scheme suggested by [9]. Once an ICN node is enabled to perform in-network transcoding, it only keeps the highest bitrate version of each segment. In this way, if an ICN router receives a request for a lower bitrate version, it will locally transcode the highest bitrate version into the requested one. The objective of this scheme is to maximize the cache hit ratio by making the limited cache space for more segments. Finally, we propose PAINT, where each ICN router strategically caches all the bitrate version for a few top popular segments, and keeps only the highest bitrate version for others. Such decision is subject to the cache capacity constraint.

Table I presents a comparison of those three strategies. In fact, the all bitrate caching scheme and the pure transcoding scheme stand for two extreme cases of PAINT. Specifically, the all rate caching scheme does not locally transcode any segment, leading to the lowest cache hit ratio and highest bandwidth cost. In contrast, the pure transcoding scheme transcodes all the segments, aiming to maximize the cache hit ratio and minimize the bandwidth cost. However, it could also result in significant transcoding cost. Therefore, PAINT is proposed as a hybrid method to balance this trade-off and optimize the total cost.

Figure 2 illustrates how an ICN node caches video segments in its local storage under PAINT. In particular, we define a general processing model by introducing a parameter $x \in [0, C/B_{sum}]$, where C is the cache capacity of an ICN node, and B_{sum} is the total size of all bitrate versions of one video segment. Under this configuration, all bitrate versions of the x most popular segments, and only the highest version of the segments, whose popularity rank is between x and h, will be cached. In this case, the amount of different segments held by this node is,

$$h = x + (C - xB_{sum})/B_h, \tag{1}$$

where B_h is the size of the highest bitrate version. We can find that when $x = C/B_{sum}$, it is the all bitrate caching case. When $x = 0$, it is the pure transcoding scheme. And when $x \in (0, C/B_{sum})$, it is the partial transcoding scheme.

In practice, given a configuration x, PAINT can be achieved in a similar way to Adaptive Replacement Cache (ARC) [17]. Specifically, each router keeps track of the requested frequency for each segment (i.e., the aggregated frequency from all bitrate versions), and maintains a LFU queue. In this way, once the router successfully forwards a data packet, it first adjusts the frequency of the requested segment, and decides whether this data should be cached locally according to its updated frequency. If it is less than the least popular one at the tail of the queue, no further operation is needed. If it is higher than the least popular one in the queue but less than the x-th popular one, only the highest bitrate version of this segment will be cached, and the least popular one will be replaced. If it is higher than the x-th popular one, the requested bitrate version will be cached, and the same bitrate version of the x-th popular will be replaced.

C. Revised ICN Routing Scheme

Figure 3 presents the packet processing engine in an ICN node under PAINT. Specifically, Figure 3(a) shows the work-flow to process the interest packets (i.e., the request) in the control plane. And Figure 3(b) shows the one for the data packets (i.e., the response) in the data plane.

There are three steps to process an interest packet. First, the router checks if the requested content can be served locally. If it is true, the router returns the content either directly from the Content Store (CS) or based on the local transcoding, depending on whether it holds the requested version in the cache. Second, if the interest cannot be served locally, the router further checks its Pending Interest Table (PIT), which keeps the records of unserved interest. If there exists the same record, it only adds the incoming interface to this record. Finally, if there is a PIT miss, the router forwards the interest via the outing interface(s) according to its Forwarding Information Base (FIB). Note, we assume a "filename-time-encoding" format to name each segment. Thus it is easy to

TABLE II: Notation table

Symbol	Definition
X	Configuration set of partial transcoding approach, (i.e., node v_k caches all-bitrates for top x_k segments).
h_i	Number of different segments can be held by node v_i.
$G(V,E)$	ICN topology with node set V and edge set E.
n	Number of core routers in G (i.e., $n = \|V\|$).
C	Cache space of each ICN router.
R	Number of streaming bitrate version.
M	Number of different video segments.
s_i^j	The i-th popular segment with the j-th highest bitrate.
B_h	Size of the highest bitrate version.
B_{sum}	Total size of all bitrate versions of one segment.
B_m	Mean size of all bitrate versions of one segment.
$B(s_i^j)$	Segment size of the i-th largest bitrate.
$P(s_i^j)$	Prob. of requesting s_i^j.
$C_{tr}^k(X)$	Transcoding cost to serve per interest from node v_k when X is configured.
$C_{ba}^k(X)$	Bandwidth cost to serve per interest from node v_k when X is configured.
C_{tot}^k	Sum of transcoding and bandwidth cost at node v_k.
$P_{tr}(x_k)$	Prob. of local transcoding ratio at node v_k.
$P_{miss}(x_k)$	Prob. of local cache miss ratio at node v_k.
P_s^k	Prob. of serving interests from v_k by origin server.
D_k	Average traversed hops to serve local cache missed interests originated from node v_k.

identify whether a requested segment is another representation of a cached copy, by longest-prefix match.

There are two steps to process a data packet. First, the router checks its PIT. If there is no matching entry, the data are unsolicited, which will be directly discarded. Second, if there is a PIT match, the data will be forwarded all the way back to the original requester. At the same time, the router deletes this PIT record, and decides whether to cache the data to local CS according to the configuration of partial transcoding method.

Note that each interest will be eventually served by either the router holding the data or the origin server. Specifically, the origin server has to serve the requested segment, if no ICN node has a copy. Otherwise, the interest packet is served by the one with the shortest response time (i.e., the duration from the moment that an user sends the interest, to another moment that this user receives the requested data), by default. It is also possible to adopt other simple yet practical schemes. For example, each node only serves those local cache hit requests, and forwards other requests directly to the origin server.

IV. Models & Problem Formulation

In this section, we present three system models and formulate the cost optimal problem for PAINT as a constrained optimization problem. For clarity in the discussion, we summarize the important notations in table II.

A. System Models

1) Topology Model: We model the ICN backbone topology as a undirected graph $G = (V, E)$, where $V = \{v_1, ..., v_n\}$ is the set of homogeneous core routers, and E denotes the set of network links between those routers. For simplicity, we consider there is only one origin content source $S \in V$.

This topology $G(V, E)$ is built from a set of resources. Specifically, each node $v \in V$ is associated with both caching and transcoding resources, and each link e is associated with bandwidth resources. The consumption on each resource component will incur a corresponding amount of cost.

2) Content Model: We consider M different video segments to be distributed over the whole network, where each segment has R different bitrate versions ranging from the smallest bitrate B_l to the highest bitrate B_h. We assume the length of each segment is the same on average, and all those R bitrates are frequently accessed (i.e., we do not consider the versions that are seldom used). Note this work only considers an adaptive video streaming application where R must be larger than 1, otherwise it is non-adaptive.

We assume the aggregated user requests towards those M different segments follow the Zipf-like distribution [18], [19], and R different bitrates follow the uniform distribution [9]. We also adopt the assumption from [15], [20], that the interests arrive at each ICN node (i.e., the edge nodes of ICN) independently (i.e., independent reference model [21]), and their arrival patterns all follow the same Possion process [22], [23]. Thus, the probability of requesting the j-th bitrate version of the i-th popular segment s_i^j at an ICN node is,

$$P(s_i^j) = \frac{1/i^\alpha}{R\sum_{k=1}^{M}(1/k^\alpha)} = \frac{1/i^\alpha}{RH_{M,\alpha}}, i = 1, ..., M, \quad (2)$$

where $H_{M,\alpha} = \sum_{l=1}^{M}(1/l^\alpha)$ is the M-th generalized harmonic number, and α is the shape parameter of Zipf distribution (α must be positive). A large α indicates more requests on popular ones and less requests on unpopular ones. Typically, α is between 0.5 and 1.5, (e.g., [13] found α is around 1). Note that, we assume this content model (i.e., $P(s_i^j)$) is independent of the configuration (i.e., x) in PAINT.

3) Cost Model: In supporting the adaptive video streaming service in ICN with in-network transcoding feature, the system would incur three different costs, including a transcoding cost, a bandwidth cost, and a caching cost. The transcoding cost is incurred when the high quality version is transcoded into a lower bitrate version. The bandwidth cost is incurred when the streaming data are transmitted from the source (i.e., either the ICN router holding the cached data or the origin server) to the user along its delivery path. The caching cost is charged when the routers cache video segments into its local cache storage. Note that, the caching cost is constant, because the finite cache space of each node must be fully filled by the traversed segments. Therefore, we only focus on the transcoding cost and the bandwidth cost in this work.

Once an ICN node receives an interest, there are three different cases with different cost components. First, if the exact version of requested segment is in the local cache, the node will directly serve it without incurring any cost. Second, if only the highest version of the requested segment is in the local cache, and it is not the requested one, the node will serve the requested version by incurring a local transcoding cost. Finally, if the requested segment is not in the local cache, additional bandwidth cost and the possible

transcoding cost on other nodes will be charged. Therefore the expectation of normalized transcoding cost by serving every interest originated from node v_k is,

$$\mathbb{E}[C_{tr}^k(X)] = w_{tr}B_h(P_{tr}(x_k) + P_{miss}(x_k)(1 - P_s^k)), \quad (3)$$

where w_{tr} is the unit price to transcode video segments, $X = \{x_1, ..., x_n\}$ is the configuration set for all nodes, $P_{tr}(x_k)$ is the local transcoding ratio at node v_k with respect to local configuration x_k, $P_{miss}(x_k)$ is the local cache miss ratio, and P_s^k is the ratio that requests originated from node v_k are served by the origin server. Note, we assume the transcoding cost is linear to the bitrate size according to Windows Azure's media service pricing model [24].

The expectation of normalized bandwidth cost to serve every interest originated from node v_k is in proportion to the amount of transmitted traffic at each edge. It is given by,

$$\mathbb{E}[C_{ba}^k(X)] = w_{ba}P_{miss}(x_k)D_kB_m, \quad (4)$$

where w_{ba} is the unit price to transmit video streaming for one hop, D_k is the average traversed hop distance of serving the cache missed interest packet, and B_m is the average size of a video segment among all bitrate versions. Under our assumption that requests on different bitrates follow uniform distribution, B_m can be approximated as B_h/R.

B. Problem Formulation

Using the system models, we formulate a constrained optimization problem to minimize the combined transcoding and bandwidth costs of per request at each node, as,

$$\min_X \quad \mathbb{E}[C_{tot}^k] = \mathbb{E}[(C_{tr}^k(X)] + \mathbb{E}[C_{ba}^k(X))], \quad (5)$$

$$s.t. \quad 0 \le x_i \le \frac{C}{B_{sum}},\ i = 1, ..., n, \quad (6)$$

where the decision variable in the objective function (5) is the in-network transcoding configuration parameter $X = \{x_1, ..., x_n\}$, and the constraint (6) captures the limitation of in-network caching space of each ICN node.

V. Analytical Solutions

This section follows four steps to analyze the optimization problem in PAINT. First, we derive the local cache hit/miss ratio and transcoding ratio. Second, we sum up all cost components to show that the objective function is convex. Then, we find the optimal strategy by solving the convex optimization problem, and analyze the impacts of different parameters. Finally, we quantify the performance gain.

A. Local Performance

1) Local Cache Hit/Miss Ratio: Local cache hit ratio refers to the probability that a request is served by its entry ICN node. Under PAINT (i.e., figure 2), the local cache hit can be approximated by the probability that the incoming interest packet is towards the top h popular segments as,

$$P_{hit}(x) = \sum_{i=1}^{h(x)} \sum_{j=1}^{R} P(s_i^j) = \frac{H_{h(x),\alpha}}{H_{M,\alpha}}, \quad (7)$$

where $h(x)$ is a function of configuration parameter x given by Eq. (1), and $H_{h(x),\alpha}$ is the $h(x)$-th generalized harmonic number of order α.

By checking the existence of the first-order derivative and the negativity of the second-order derivative of $P_{hit}(x)$ with respect to x, we obtain the following lemma.

Lemma 1: Local cache hit ratio is a concave function in terms of its local configuration setting x.

Proof: See Appendix A for a completed proof. ■

We can also easily have the cache miss ratio at node v_k as,

$$P_{miss}(x_k) = 1 - P_{hit}(x_k). \quad (8)$$

It is also easy to find that the local cache miss ratio is a convex function of x_k, because $P_{hit}(x_k)$ is concave.

2) Local Transcoding Ratio: The local transcoding ratio refers to the probability that the entry ICN router served its local interest packets after on-line transcoding. Specifically, the local transcoding ratio can be approximated by the probability that the popularity rank of the requested segment is between x and h. Thus, it is given by,

$$P_{tr}(x) = \sum_{i=x+1}^{h(x)} \sum_{j=2}^{R} P(s_i^j) = \frac{(R-1)(H_{h(x),\alpha} - H_{x,\alpha})}{RH_{M,\alpha}}. \quad (9)$$

B. Bounds of Total Cost Function

In order to ease the analysis and derive meaningful results, we characterize an upper and lower bound of the total cost function, instead of its exact form. In particular, we consider two extreme cases as follows.

First, there is an upper bound of the total cost, when all the local cache misses are served by the origin server. Specifically, ICN aims to shift the content from the origin server to a place that is closer to the user, so that the operational cost can be reduced. As a result, it is the last choice to serve the requests by the origin server, only if no other solution yields lower cost. In this case, $P_s^k = 1$ and D_k equals to the shortest hop distance from node v_k to the origin server. Thus, the total cost upper bound of serving the requests from node v_k is,

$$\overline{C_{tot}^k(x_k)} = w_{tr}B_hP_{tr}(x_k) + w_{ba}B_mD_kP_{miss}(x_k). \quad (10)$$

On the contrary, there is a lower bound when all those cache misses are served by the neighboring nodes without transcoding, instead of the origin server (i.e., $P_s^k = 0$). Note this is the most ideal case, which may not really exist in practise. In this way, we have the total cost to serve the requests from node v_k as,

$$\underline{C_{tot}^k(x_k)} = w_{tr}B_hP_{tr}(x_k) + w_{ba}B_mP_{miss}(x_k). \quad (11)$$

Finally we have the following theorem, which allows us to apply the convex optimization method to solve the problem.

Theorem 1: The total cost function is convex in terms of its local configuration setting x.

Proof: See Appendix B for a completed proof. ■

C. Optimal Strategy

We next derive the optimal strategy for PAINT. The objective is to obtain the closed-form solution and analyze the impacts of different system parameters. For simplicity, we only focus on the cost upper bound (i.e., Eq. (10)) and its strategy, as a feasible solution. In this case, the total cost at each node only depends on its local configuration. As a result, the cost of the whole network can be minimized as long as each node independently achieves its optimal strategy.

Theorem 2: The optimal partial in-network transcoding configuration at node v_k is,

$$x_k^\star = \min\{\frac{C}{B_h(\eta_1^{1/\alpha}+b)}, \frac{C}{B_{sum}}\}, \tag{12}$$

where $b = \frac{B_{sum}}{B_h} - 1$, and $\eta_1 = b(\frac{w_{ba}B_m D_k R}{w_{tr}B_h(R-1)} - 1)$.

Proof: Let $\frac{\partial C_{tot}(x)}{\partial x} = 0$, we have,

$$x^{-\alpha} = \eta_1(c - bx)^{-\alpha},\ \alpha \neq 1, \tag{13}$$

where we import b, c, and η_1 to simplify the expression. By solving this equation, we have the optimal strategy $x_k^\star = \frac{c}{\eta_1^{1/\alpha}+b}$. In addition, it is also necessary to make sure the generated solution does not violate the cache capacity constraint (6). Thus, we obtain the final solution as Eq. (12).

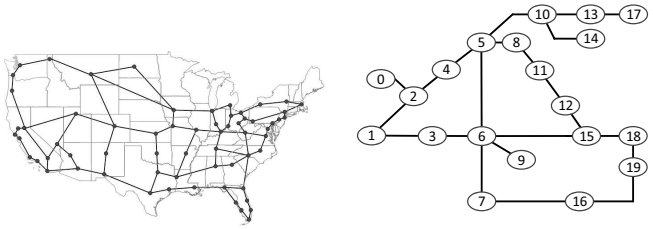
Note when $\alpha = 1$, there is no optimal solution. However, we can simply neglect this factor in practice, because the Zipf distribution parameter α could be very close to 1, but not necessarily equals to 1. ■

Based on the closed form of optimal strategy, we analytically investigate how those parameters affect the optimal decision. First, as the cache capacity C increases, $x^\star$ increases linearly, but $\frac{x^\star}{C}$ remains the same. Note that, such statement is valid under the assumption that the cache space of one ICN node is never large enough to keep all segments (i.e., $h(x^\star) < M$ holds). This implies the optimal strategy in PAINT is not affected by the cache space size in typical ICN environment. Second, as D_k increases, $x^\star$ decreases accordingly. This means the closer the ICN node is to the origin server, the more cache space should be allocated for all-bitrate caching. Finally, as $\frac{w_{ba}}{w_{tr}}$ increases, $x^\star$ decreases accordingly. This means that if the transcoding cost is high and the bandwidth cost is low, it is advantageous to cache all-bitrate versions for more segments. We will further evaluate these insights in detail in Section VI.

D. Cost Saving

We quantity the cost savings by comparing the cost upper bound of PAINT (i.e., $x_k^\star$) with the all rate caching scheme [3] (i.e., $x_k^c = h_k^c = \frac{C}{B_{sum}}$) and the pure transcoding scheme [9] (i.e., $x_k^p = 0$, and $h_k^p = \frac{C}{B_h}$). In particular, we substitute those values back to Eq. (10), and have the total cost of all rate caching scheme as,

$$C_{tot}^c = \frac{w_{ba}D_k B_m(H_{M,\alpha} - H_{h_k^c,\alpha})}{H_{M,\alpha}}, \tag{14}$$



(a) DAPRA's CORONET (b) CERNET2

Fig. 4: Real-world network topologies

TABLE III: Topological parameters

Topology	$\|V\|$	$\|E\|$	Radius	Diameter
CORONET	64	166	10 hops	16 hops
CERNET2	20	22	4 hops	7 hops

the total cost of pure transcoding scheme as,

$$C_{tot}^p = \frac{w_{ba}D_k B_m(H_{M,\alpha} - H_{h_k^p,\alpha})}{H_{M,\alpha}} + \frac{w_{tr}B_h H_{h_k^p,\alpha}(R-1)}{H_{M,\alpha}R}, \tag{15}$$

and the total cost of PAINT as,

$$C_{tot}^\star = \frac{w_{ba}D_k B_m(H_{M,\alpha} - H_{h_k^\star,\alpha})}{H_{M,\alpha}} + \frac{w_{tr}B_h(H_{h_k^\star,\alpha} - H_{x_k^\star,\alpha})(R-1)}{H_{M,\alpha}R}. \tag{16}$$

Thus the cost saving is $C_{tot}^c - C_{tot}^\star$ for all rate caching scheme, and $C_{tot}^p - C_{tot}^\star$ for pure transcoding scheme.

We have a few insights from those equations. First, PAINT degrades to the all rate caching scheme, if $D_k \leq \frac{w_{tr}B_h(R-1)}{w_{ba}B_m R}$ (i.e., $C_{tot}^\star = C_{tot}^c$) holds. This implies it is better to disable the in-network transcoding for those nodes closed to the origin server (i.e., within a distance threshold, which increases linearly as the transcoding unit cost increases or the bandwidth unit cost decreases). Second, when $\alpha > 0$, $w_{tr} > 0$, the pure transcoding scheme yields higher cost than PAINT, under our previous assumption that the requests on different bitrate versions follow uniform distribution,, because both $x_k^\star > 0$ and $C_{tot}^p - C_{tot}^\star > 0$ hold. This means the pure transcoding scheme is not cost-efficient enough in most cases. Finally, we find even the cost upper bound of PAINT saves significant costs in most cases. This proves the efficiency of PAINT. We will discuss these insights in detail in Section VI.

VI. Performance Evaluation

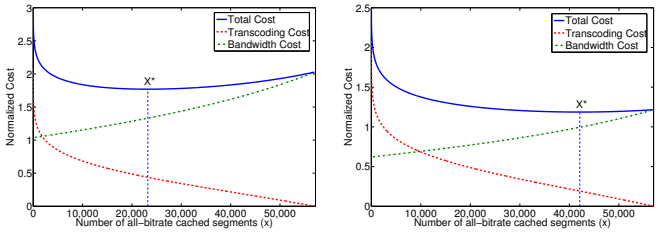
This section numerically evaluates PAINT by using real-world adaptive streaming settings and real network topologies.

A. Experimental Setup

We use two real network topologies as shown in Figure 4, including DAPRA's CORe Optical NETworks (CORONET) in the U.S [25], and China Education and Research NETwork 2 (CERNET2) [26]. Table III summarizes important metrics of these two networks. Specifically, the node number and the edge number are the basic property of any topology. Besides, we also focus on their radius and diameter, because they stand

TABLE IV: Bitrate levels in real adaptive streaming system

Type	360p	480p	540p	720p	960p
Bitrate	0.4Mbps	0.6Mbps	0.9Mbps	1.2Mbps	1.5Mbps



(a) Cost at Miami node if origin server is at Boston in CORONET (b) Cost at Xiamen node if origin server is at Tianjing in CERNET2

Fig. 5: Numerical results of the costs and the optimal strategy

for a lower bound and an upper bound of the longest shortest hop distance from one node to any other node, respectively. In particular, for a graph $G = (V, E)$, its radius is defined as $\min_{u \in V}(\max_{v \in V} d(u, v))$, and its diameter is defined as $\max_{u,v \in V} d(u, v)$, where $d(u, v)$ is the shortest hop distance between node u and v.

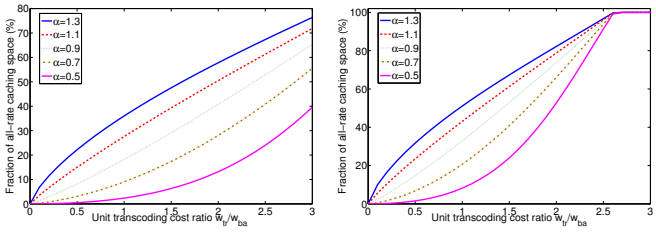
Table IV lists all the bitrate levels of a content in a real adaptive streaming system [27]. Based on these settings, we have $R = 5$, $B_h = 1.5Mb$, $B_{sum} = 4.6Mb$, $B_m = 0.775Mb$. And we assume there are $M = 500,000$ different segments.

B. Optimal Configuration

This subsection first verifies our analytical solution by comparing it with numerical results. Then we evaluate the optimal strategy under different experimental settings. Our focus is to understand the impacts of various system parameters, ultimately obtaining operational guidelines for deployment.

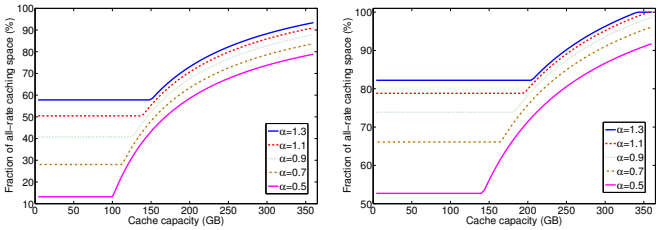
1) Verification: Figure 5 demonstrates the numerical solutions of two examples, where the results are generated in an exhaustive manner. In particular, Figure 5(a) shows the total cost and the optimal strategy for the node at Miami if the origin is at Boston (i.e., $D_k = 10$ based on CORONET). And Figure 5(b) presents the same metrics for the node at Xiamen, if the origin server is at Tianjing (i.e., $D_k = 6$ based on CERNET2). For other parameter settings, we normalize the bandwidth cost as $w_{ba} = 1$, and set $w_{tr} = 2$, $\alpha = 0.9$, and $C = 32GB$. The impacts of these parameters on the optimal strategy will be investigated in the following subsections.

This experiment confirms our analytical solutions. Specifically, first, we find the numerical solution (e.g., $x_1^{\star\prime} = 23220$ in Figure 5(a) and $x_2^{\star\prime}$=42090 in Figure 5(b)) fits well with our analytical result (e.g., $x_1^{\star}$=23221 and $x_2^{\star}$=42092 according to Eq. (12)). Second, there is a clear trade-off between the transcoding and bandwidth costs in both figures. As x increases, the transcoding cost decreases and the bandwidth cost increases. This verifies our analysis in Section III-B. Finally, we observe the total cost is a convex function that for any $x_1, x_2 \in [0, C/B_{sum}]$, we always have $c_{tot}(\frac{x_1+x_2}{2}) \leq \frac{c_{tot}(x_1)+c_{tot}(x_2)}{2}$. This verifies our analysis in Section V-C.



(a) Optimal strategy at Miami node if origin server is at Boston (b) Optimal strategy at Xiamen node if origin server is at Tianjing

Fig. 6: Optimal strategy vs. unit transcoding cost ratio



(a) Optimal strategy at Miami node if origin server is at Boston (b) Optimal strategy at Xiamen node if origin server is at Tianjing

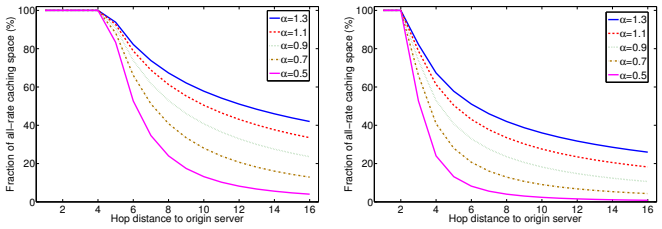
Fig. 7: Optimal strategy vs. cache capacity

2) Impact of unit transcoding cost ratio $\frac{w_{tr}}{w_{ba}}$: Figure 6 presents the relationship between the unit transcoding price ratio w_{tr}/w_{ba} and the optimal fraction of all rate caching space $\frac{B_{sum}x^{\star}}{C}$, where the cache capacity is set at $C = 32GB$.

We have a few observations from this experiment. First, more cache space should be allocated to keep all-bitrate versions for more segments, as the transcoding cost ratio increases. This suggests us to forward more requests to the origin server rather than serving them locally based on real-time transcoding, when the transcoding cost is high. Second, when the unit transcoding cost is very high, PAINT degrades to the all rate caching scheme by allocating all the space to cache all rate versions, as we discussed in Section V-D. Finally, for a fixed price ratio w_{tr}/w_{ba}, as the Zipf parameter α increases, the fraction of all rate caching space increases as well. In this case, a large Zipf parameter indicates there are relatively more requests on very popular segments, and fewer requests on the unpopular ones. Thus, it is better to cut the transcoding cost for the relatively popular segments and increase the bandwidth cost for the relatively unpopular ones by setting a larger $x^{\star}$. This observation also applies for Figure 7 and Figure 8.

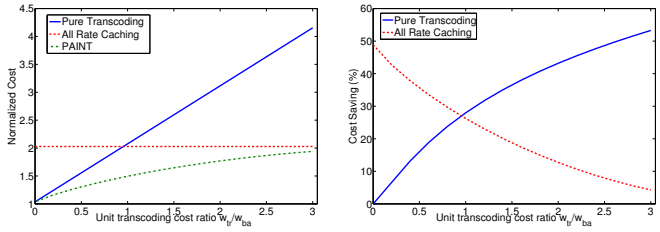
3) Impact of cache capacity C: Figure 7 shows the relationship between the cache capacity C and the optimal fraction of all rate caching space $\frac{B_{sum}x^{\star}}{C}$, where the unit transcoding price ratio is set as $w_{tr}/w_{ba} = 2$.

We make the following observations. First, when cache capacity C is small (e.g., $C <$ 120GB for $\alpha = 0.7$), as the cache capacity C increases, the fraction remains the same. This is because more cache space leads to the improvements on both local cache hit ratio and transcoding ratio, but it does not affect the trade-off between the transcoding and bandwidth



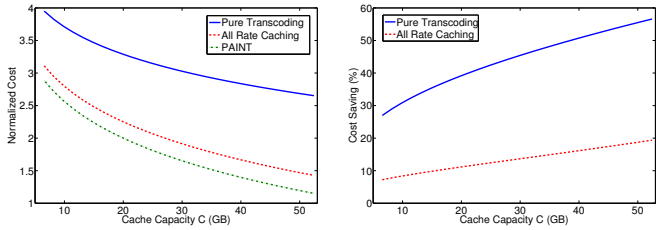
(a) $w_{tr}/w_{ba} = 2$ (b) $w_{tr}/w_{ba} = 1$

Fig. 8: Optimal strategy vs. hop distances to origin server



(a) Total Cost (b) Cost Saving Percentage

Fig. 9: Total cost vs. unit transcoding cost ratio



(a) Total Cost (b) Cost Saving Percentage

Fig. 10: Total cost vs. cache capacity

costs. This verifies our analysis in Section V-C. Second, a larger Zipf parameter α leads to a higher fraction of all rate caching space. This confirms our findings from Figure 6. Finally, we note when cache capacity C is sufficiently large to keep all segments (e.g., $C > 120$GB for $\alpha = 0.7$), the fraction increases because more bitrate versions of those popular segments can be kept in the cache. But this case is not practical in reality, because the cache space of an ICN node is usually much smaller than the total volume of content.

4) Impact of hop distance to origin server D_k: Figure 8 presents the relationship between the hop distance from a node to the origin server and the optimal fraction of all rate caching space $\frac{B_{sum}x^\star}{C}$ on this node, where the unit transcoding price ratio is $w_{tr}/w_{ba} = 2$ and the cache capacity is $C = 32GB$.

This experiment reveals the following insights. First, the all rate caching scheme stands for the optimal strategy when D_k is less than a threshold $\tilde{D_k}$ (e.g., $D_k \leq 4$ when $w_{tr}/w_{ba} = 2$). In this case, the local transcoding cost on a relatively unpopular content become higher than the bandwidth cost to retrieve a relatively popular one from the origin server. Thus it is optimal to disable in-network transcoding. Second, the threshold $\tilde{D_k}$ is affected by the price ratio w_{tr}/w_{ba}, but not by the Zipf parameter α (e.g., $D_k = 4$ holds for different α in Figure 8(a)). This verifies our analysis in Section V-D. Finally, the fraction decreases in a log scale, as the hop distance D_k increases. This can be traced back to the logarithm nature of Zipf law.

C. Cost Savings

This subsection numerically evaluates the cost savings and checks how those system parameters affect the performance.

1) Impact of unit transcoding cost ratio $\frac{w_{tr}}{w_{ba}}$: Figure 9 presents the optimal total cost and cost savings compared with pure transcoding and all rate caching scheme, with respect to the unit transcoding cost ratio $\frac{w_{tr}}{w_{ba}}$. Here we set $\alpha = 0.9$, $C = 32GB$ and $D_k = 10$.

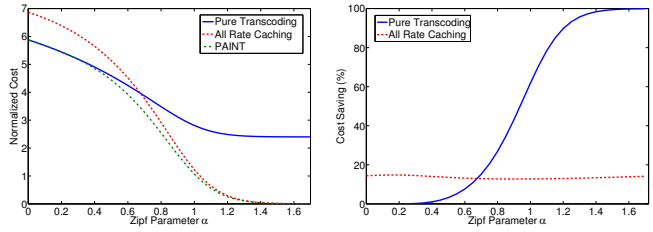
We report the following observations. First, as w_{tr} increases, the total cost of pure transcoding scheme increases linearly but the one of the all rate caching scheme remains the same. This can be explained as follows. For pure transcoding scheme, the amount of transcoded segments does not change, so the total cost is in proportional to the unit transcoding cost price w_{tr}. For the all rate caching scheme, there is no transcoding at all, so its total cost is not related to the transcoding price. Second, PAINT saves significant cost (i.e., up to 50% in this experiment) compared to existing schemes. Third, as w_{tr} increases, the cost overhead of pure transcoding scheme increases and the one of all rate caching scheme decreases. Finally, we notice that, PAINT successfully finds the convex hull between those two methods, with respect to w_{tr}/w_{ba}.

2) Impact of cache capacity C: Figure 10 shows the optimal total cost and cost savings, in terms of the cache capacity C, where $\alpha = 0.9$, $\frac{w_{tr}}{w_{ba}} = 2$ and $D_k = 10$. Here we only consider the general case, that the cache space of an ICN node is much smaller than the total volume of content.

This experiment reveals the total cost of all schemes goes down, and cost savings compared to both existing schemes goes up, as the cache capacity increases. Specifically, this is because the ICN node keeps more content, when the cache space becomes larger. As a result, more requests can be served locally without incurring the additional bandwidth cost, and the total cost gets improved. At the same time, given fixed video segment size, a large cache space also allows us to achieve more fine-grained scheduling to further improve the performance. Therefore, the cost savings also grow accordingly (e.g., the saving compared with all rate caching scheme increases from 8% to 20% when the cache space grows from 4GB to 52GB). This suggests that PAINT works better for the ICN nodes with relatively large cache space.

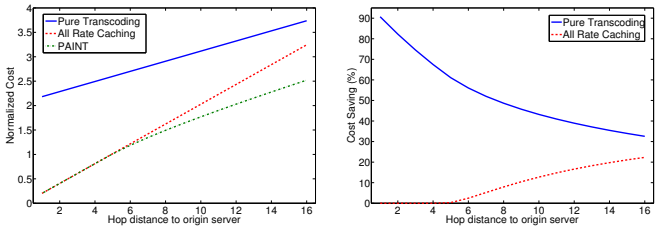
3) Impact of Zipf parameter α: Figure 11 shows the optimal total cost and the savings at an ICN node, with respect to the Zipf parameter α of the users' request patterns, where $C = 32GB$, $\frac{w_{tr}}{w_{ba}} = 2$ and $D_k = 10$.

We have the following observations from this set of experiment. First, the total cost of pure transcoding scheme tends to be the optimal one, when α tends to zero. In this case, the user requests on each segment tends to follow an uniform distribution. Thus, the benefits of caching popular segments



(a) Total Cost (b) Cost Saving Percentage

Fig. 11: Total cost vs. Zipf parameter α



(a) Total Cost (b) Cost Saving Percentage

Fig. 12: Total cost vs. hop distances to origin server

are diminished, and the optimal strategy is to maximize the local cache hit ratio by only caching the highest quality version for each segment. However, on the other side, when α tends to be larger (e.g., 1.6 in this case), the cost overhead of pure transcoding scheme is several times of the optimal one generated by PAINT. This is because the cache hit ratio cannot be improved by simply caching more segments, when the request pattern is heavily long-tailed. In this case, it is better to focus more on those very popular segments. Second, the cost overhead of all rate caching scheme keeps almost unchanged, as α increases. This can be attributed to the fact that, the local cache hit ratio of both all rate caching scheme and PAINT increases at the same scale. Finally, we again note that, PAINT successfully finds the convex hull of two existing schemes, with respect to the Zipf parameter α.

4) Impact of hop distance to origin server D_k: Figure 12 presents the optimal total cost and the savings at an ICN node, in terms of its hop distance to the origin server D_k, where $C = 32GB$, $\alpha = 0.9$ and $\frac{w_{tr}}{w_{ba}} = 2$.

The observations on this set of experiment are as follows. First, the total cost of all three schemes increases as the hop distance grows, because of the increment on the bandwidth cost to serve those cache missed segments. Second, when hop distance is short (e.g., $D_k \leq 4$ in this case), all rate caching scheme stands for the optimal strategy. This matches our analysis for Figure 8. Finally, the cost saving compared with all rate caching scheme increases, whereas the one compared with pure transcoding scheme decreases, as the hop distance grows. To understand this phenomenon, we look at the processing model of the two existing schemes as shown in Table I. Specifically, the all rate caching scheme has the lowest local cache hit ratio that maximizes the amount of cache missed requests, while the pure transcoding scheme operates in an opposite way to minimize the cache missed requests. Besides, as the hop distance increases, the unit bandwidth cost to serve per segment increases. Therefore, the gap between the total cost of these two schemes gets smaller.

VII. Conclusion and Future Works

In this paper, we propose PAINT (PArtial In-Network Transcoding) scheme to optimize the operational cost of delivering adaptive video streaming over ICN. In particular, we consider both in-network caching and transcoding features, and formulate an optimization problem to examine the trade-off between the transcoding and bandwidth costs. Then, we analytically derive the optimal PAINT strategy on provisioning storage and transcoding resources at each ICN node, and quantify the cost savings compared with existing schemes. Finally, we verify our solution based on intensive numerical evaluations. The results indicate significant cost savings (e.g., up to 50% in typical scenarios) can be achieved by PAINT. In addition, the optimal strategy and the cost savings can be affected by the cache capacity, the unit price ratio, the hop distance to origin server, and the Zipf parameter of users' request patterns. These insights provide operational guidelines to the design for the future Internet.

Our future work will cover the following aspects. First, we will consider the cooperative partial in-network transcoding scheme in ICN, so that the total cost over the whole network can be further reduced. Second, we plan to generalize our model to capture the popularity dynamics of video segments. Finally, we are interested in integrating our in-network transcoding function into open source ICN project (e.g., CCNx) and perform Internet-based evaluations.

Appendix A
Proof of Lemma 1

In this section, we accomplish the proof of lemma 1.

First, we expand the generalized harmonic number to obtain a more clear form of Eq. (7). Specifically, when $\alpha = 1$, it is a special case that $H_{h(x),\alpha}$ become a harmonic number as [28],

$$H_{h(x),1} \approx \ln h(x) + \gamma, \tag{17}$$

where $\gamma \approx 0.577$ is the Euler-Mascheroni constant. When $\alpha \neq 1$, we consider its series expansion at infinity as,

$$H_{h(x),\alpha} \approx \frac{h(x)^{1-\alpha}}{1-\alpha} + \zeta(\alpha), \ \alpha \neq 1, \tag{18}$$

where $\zeta(\alpha)$ is Riemann Zeta function, which is a constant for a specific α. As a result, we organize local cache hit ratio into,

$$P_{hit}(x) \approx \begin{cases} (\frac{h(x)^{1-\alpha}}{1-\alpha} + \zeta(\alpha))\frac{1}{H_{M,\alpha}}, & \alpha \neq 1 \\ \frac{\gamma + \ln h(x)}{H_{M,\alpha}}, & \alpha = 1 \end{cases}, \tag{19}$$

where $H_{M,\alpha}$ is constant which is independent with x.

Second, we derive both the first and second-order derivative to check the concavity of local cache hit function. Specifically, the first-order derivative is,

$$\frac{\partial P_{hit}(x)}{\partial x} = \begin{cases} \frac{(-b)(c-bx)^{-\alpha}}{a}, & \alpha \neq 1 \\ \frac{b}{a(bx-c)}, & \alpha = 1 \end{cases}, \tag{20}$$

where $a = H_{M,\alpha}$, $b = \frac{B_{sum}}{B_h} - 1$, and $c = C/B_h$ are all positive constants that are independent of x.

And the second-order derivative is,

$$\frac{\partial^2 P_{hit}(x)}{\partial x^2} = \begin{cases} -\frac{\alpha b^2 (c-bx)^{-\alpha-1}}{a}, & \alpha \neq 1 \\ -\frac{b^2}{a(bx-c)^2}, & \alpha = 1 \end{cases}. \quad (21)$$

Finally we prove the cache hit function is concave of x by showing its second-order derivative is always negative. Specifically, when $\alpha = 1$, it is obviously that $(b-1)^2 > 0$ and $a(bx-c)^2 > 0$. Thus $\frac{\partial^2 P_{hit}(x)}{\partial x^2} < 0$ holds. When $\alpha \neq 1$, we find $c - bx = h(x) > 0$ and $h(x)^{-\alpha-1} > 0$. Therefore $\frac{\partial^2 P_{hit}(x)}{\partial x^2} < 0$ still holds.

In summary, we conclude that the local cache hit ratio is a concave function with respect to x.

APPENDIX B
PROOF OF THEOREM 1

In this section, we provide the completed proof of theorem 1 by adopting the same approach as in Appendix A.

First, we normalize the total cost function by dividing it by $\frac{w_{tr} B_h (R-1)}{R}$, and importing an extra parameter $\eta = \frac{w_{ba} B_m D_k R}{w_{tr} B_h (R-1)}$ by capturing the scale of $P_{miss}(x)$ in the upper (i.e., Eq. (10)) and lower bound (i.e., Eq. (11)). In this way, we re-organize the total cost function into,

$$C_{tot}(x) = \frac{R}{R-1} P_{tr}(x) + \eta P_{miss}(x). \quad (22)$$

Second, we substitute Eq. (8), (9), (17), and (18), into Eq. (22), and obtain,

$$C_{tot}(x) \approx \begin{cases} \frac{(1-\eta)h(x)^{1-\alpha} - x^{1-\alpha} + \eta(1-\zeta(\alpha))}{H_{M,\alpha}(1-\alpha)}, & \alpha \neq 1 \\ \frac{(1-\eta)\ln h(x) - \ln x - \eta\gamma}{H_{M,\alpha}}, & \alpha = 1 \end{cases}. \quad (23)$$

Then we derive the first-order derivative of the total cost function (i.e., Eq. (23)) as,

$$\frac{\partial C_{tot}(x)}{\partial x} = \begin{cases} \frac{b(\eta-1)(c-bx)^{-\alpha} - x^{-\alpha}}{a}, & \alpha \neq 1 \\ \frac{b(1-\eta)}{a(bx-c)} - \frac{1}{ax}, & \alpha = 1 \end{cases}, \quad (24)$$

where a, b and c are exactly the same as used in Eq. (20).

And the second-order derivation of the total cost function can be derived into,

$$\frac{\partial^2 C_{tot}(x)}{\partial x^2} = \begin{cases} \frac{\alpha(b^2(\eta-1)(c-bx)^{-\alpha-1} + x^{-\alpha-1})}{a}, & \alpha \neq 1 \\ \frac{b^2(\eta-1)}{a(c-bx)^2} + \frac{1}{ax^2}, & \alpha = 1 \end{cases}. \quad (25)$$

Finally, we check the positivity of the second-order derivation (i.e., Eq. (25)) to prove the total cost function (i.e., Eq. (22)) is convex in terms of x. In particular, we have $b^2 > 0$, $\eta - 1 > 0$, $(c - bx) > 0$, and $x > 0$. When $\alpha \neq 1$, there are both $b^2(\eta-1)(c-bx)^{-\alpha-1} > 0$ and $x^{-\alpha-1} > 0$. Therefore we have $\frac{\partial^2 C_{tot}(x)}{\partial x^2} > 0$. When $\alpha = 1$, we have $a(c-bx)^2 > 0$ and $ax^2 > 0$. Thus $\frac{\partial^2 C_{tot}(x)}{\partial x^2} > 0$ still holds.

In summary, we conclude that the total cost function is convex with respect to the PAINT configuration parameter x.

REFERENCES

[1] Sandvine, "The global internet phenomena report," https://www.sandvine.com/downloads/general/global-internet-phenomena/2013/sandvine-global-internet-phenomena-report-1h-2013.pdf, 2013.

[2] Accenture, "Video-over-internet consumer survey 2013," http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Video-Over-Internet-Consumer-Survey-2013.pdf, 2013.

[3] V. Jacobson, D. K. Smetters, and et al., "Networking named content," in *ACM CoNEXT*, 2009, pp. 1–12.

[4] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, 2012.

[5] S. Lederer, C. Timmerer, C. Westphal, and C. Mueller, "Adaptive video streaming over ICN," *Internet-Drafts*, 2014.

[6] D. Kulinski and J. Burke, "NDN video: Live and prerecorded streaming over NDN," The NDN Project Team, Tech. Rep., 2012.

[7] I. Ahmad, X. Wei, Y. Sun, and Y.-Q. Zhang, "Video transcoding: an overview of various techniques and research issues," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 793–804, 2005.

[8] Y. Wang, J.-G. Kim, S.-F. Chang, and H.-M. Kim, "Utility-based video adaptation for universal multimedia access (UMA) and content-based utility function prediction for real-time video transcoding," *IEEE Transactions on Multimedia*, vol. 9, no. 2, pp. 213–220, 2007.

[9] R. Grandl, K. Su, and C. Westphal, "On the interaction of adaptive video streaming with content-centric networking," *arXiv preprint arXiv:1307.0794*, 2013.

[10] Y. Jin, T. Xie, Y. Wen, and H. Xie, "Multi-screen cloud social tv: transforming tv experience into 21st century," in *Proceedings of ACM international conference on Multimedia*, 2013, pp. 435–436.

[11] Y. Jin, Y. Wen, and H. Hu, "Minimizing monetary cost via cloud clone migration in multi-screen cloud social tv system," in *IEEE GLOBECOM*, 2013.

[12] H. Zhang, Y. Jin, W. Zhang, and Y. Wen, "Enhancing user experience for multi-screen social tv streaming over wireless networks," in *IEEE GLOBECOM (submitted)*, 2014.

[13] S. K. Fayazbakhsh, Y. Lin, and et al., "Less pain, most of the gain: Incrementally deployable icn," in *ACM SIGCOMM*, 2013, pp. 147–158.

[14] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache less for more in information-centric networks," in *IFIP Networking*, 2012, pp. 27–40.

[15] Y. Li, H. Xie, Y. Wen, and Z.-L. Zhang, "Coordinating in-network caching in content-centric networks: Model and analysis," in *IEEE International Conference on Distributed Computing Systems*, 2013.

[16] D. Rossi and G. Rossini, "On sizing ccn content stores by exploiting topological information," in *IEEE INFOCOM WKSHPS*, 2012.

[17] N. Megiddo and D. S. Modha, "ARC: A self-tuning, low overhead replacement cache." in *USENIX FAST*, vol. 3, 2003, pp. 115–130.

[18] L. Guo, S. Chen, and X. Zhang, "Design and evaluation of a scalable and reliable P2P assisted proxy for on-demand streaming media delivery," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 5, pp. 669–682, 2006.

[19] W.-P. Yiu, X. Jin, and S.-H. Chan, "VMesh: Distributed segment storage for peer-to-peer interactive video streaming," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1717–1731, 2007.

[20] S. Guo, H. Xie, and G. Shi, "Collaborative forwarding and caching in content centric networks," in *IFIP Networking*, 2012, pp. 41–55.

[21] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *IEEE INFOCOM*, vol. 1, 1999, pp. 126–134.

[22] S. Jin and A. Bestavros, "GISMO: a generator of internet streaming media objects and workloads," *ACM SIGMETRICS Performance Evaluation Review*, vol. 29, no. 3, pp. 2–10, 2001.

[23] H. Zhang, Z. Zhang, H. Dai, and S. Chen, "Packet spreading without relaying in mobile wireless networks," in *IEEE WCSP*, 2012, pp. 1–6.

[24] "Windows azure media services pricing details," http://www.windowsazure.com/en-us/pricing/details/media-services/.

[25] A. Saleh, "Dynamic multi-terabit core optical networks: architecture, protocols, control and management (coronet)," *DARPA BAA*, 2006.

[26] "CERNET2 topology," http://www.edu.cn/20060111/3170220.shtml.

[27] S. Lederer, C. Mueller, and et al., "Distributed dash dataset," in *ACM MMsys*, 2013, pp. 131–135.

[28] J. Havil, "Gamma: exploring eulers constant," *The Australian Mathematical Society*, p. 250, 2003.