# Orchestrating Caching, Transcoding and Request Routing for Adaptive Video Streaming Over ICN

HAN HU, Beijing Institute of Technology, China
YICHAO JIN and YONGGANG WEN, Nanyang Technological University, Singapore
CEDRIC WESTPHAL, University of California, Santa Cruz, USA

Information-centric networking (ICN) has been touted as a revolutionary solution for the future of the Internet, which will be dominated by video traffic. This work investigates the challenge of distributing video content of adaptive bitrate (ABR) over ICN. In particular, we use the in-network caching capability of ICN routers to serve users; in addition, with the help of named function, we enable ICN routers to transcode videos to lower-bitrate versions to improve the cache hit ratio. Mathematically, we formulate this design challenge into a constrained optimization problem, which aims to maximize the cache hit ratio for service providers and minimize the service delay for endusers. We design a two-step iterative algorithm to find the optimum. First, given a content management scheme, we minimize the service delay via optimally configuring the routing scheme. Second, we maximize the cache hits for a given routing policy. Finally, we rigorously prove its convergence. Through extensive simulations, we verify the convergence and the performance gains over other algorithms. We also find that more resources should be allocated to ICN routers with a heavier request rate, and the routing scheme favors the shortest path to schedule more traffic.

CCS Concepts: • **Information systems** → **Multimedia streaming**; • **Networks** → **In-network processing**; *Middle boxes/network appliances*; Naming and addressing;

Additional Key Words and Phrases: Information Centric Networking (ICN), adaptive video streaming, partial caching, video transcoding

**ACM Reference format:**
Han Hu, Yichao Jin, Yonggang Wen, and Cedric Westphal. 2019. Orchestrating Caching, Transcoding and Request Routing for Adaptive Video Streaming Over ICN. *ACM Trans. Multimedia Comput. Commun. Appl.* 15, 1, Article 24 (January 2019), 23 pages.
https://doi.org/10.1145/3289184

## 1 INTRODUCTION

Recent years have witnessed the exploding growth of video traffic from various end devices. As predicted by Cisco [5], global IP video traffic will grow threefold from 2016 to 2021, accounting for

This work was done when Han Hu was a research fellow with Nanyang Technological University.
Authors' addresses: H. Hu, Beijing Institute of Technology, School of Information and Electronics, No. 5 South zhongguancun Street, Haidian District, Beijing 100008, P. R. China; email: hhu@bit.edu.cn; Y. Jin and Y. Wen, Nanyang Technological University, School of Computer Science and Engineering, Blk N4-02c-95, Nanyang Avenue, Singapore 639798, Singapore; emails: {yjin3, ygwen}@ntu.edu.sg; C. Westphal, University of California, School of Computer Science and Engineering, 1156 High St., Santa Cruz, CA95064; email: cedric@soe.ucsc.edu.

1551-6857/2019/01-ART24 $15.00
https://doi.org/10.1145/3289184

82% of all consumer Internet traffic by 2020. Streaming data are adaptively delivered in different formats and bitrates to different devices, including PCs, smart TVs, smartphones, and tablets [32]. This ever-growing volume of video traffic strains current Internet infrastructure regarding bandwidth resources and causes service delays and appeals for novel network architectures to reduce operational costs and to enhance service quality [17, 18].

Information-centric Networking (ICN) [8] has emerged as a promising framework to efficiently distribute video streams over the Internet in the future. ICN enables caching of addressable content chunks in every router and replacement of uncached chunks at line speed, that is, every item of content is uniquely identified and accessed without being associated to a host address. This paradigm intrinsically supports in-network caching, allocating the replicated contents from the origin server to a location closer to users. Some large files are divided into different chunks, each of which has a corresponding ID and access frequency. These chunks may be cached into different routers within the service networks. Therefore, there may be various routing schemes with different QoS levels when accessing a specific file, and multiple applications can share the same content chunks identified by the unique names. These features offer a great opportunity for the architectural design and optimization of cost-efficient video distribution.

ICN technology to date is far from mature, especially for the distribution of adaptive media streaming. In comparison with web contents, which have been well studied over ICN [15], video contents have unique characteristics. They pose significant challenges to ICN [31]. In general, popular videos are cached at the edge ICN routers and unpopular videos can be accessed from the content source. For the cached popular videos, they are consumed by a set of heterogeneous end devices, requiring different resolutions, formats, and bitrates [6, 33, 39]. Under current ICN frameworks, different versions (e.g., formats, resolutions, and bitrates) of the same segment come with different names. Consequently, they are cached as different contents. However, different versions of the same video have an inherent relationship. For instance, the highest bitrate version can be transcoded to other versions [11, 12]. Requests for unpopular videos will incur a long hop distance from edge routers to the content source. Routing schemes — including bandwidth sharing, path planning, and so forth — should be exploited to reduce service latency.

Current efforts focus on caching schemes and cache size allocation [8] to improve ICN caching performance. In these works, contents are divided into chunks, each of which has a unique identification. Under this framework, different versions of the same video chunk are treated as different contents and, therefore, cannot benefit from transcoding. This eventually decreases cache use and results in needless network traffic with redundant contents. A pioneer work from Grandl et al. [13] presented a pure in-network transcoding scheme to store only the highestquality segment and derive others by real-time transcoding. However, this policy incurs heavy computation load at edge routers. Our recent works [22, 24, 25] combined partial caching and transcoding to balance this trade-off. As for long-tailed video requests, to the best of our knowledge, few efforts have been made. Existing ICN frameworks are designed to optimize the overall performance for network administrators. ICN can significantly reduce service latency to the cached contents (popular). One drawback of this design mechanism is that it performs worse for those uncached contents.

In this article, we aim to jointly tackle the aforementioned two challenges. First, by integrating transcoding to the named function at each ICN router, we adopt a partial caching scheme to improve the cache hit ratio. Named function extends the classic ICN to support the concept of function definition and application to data [38, 42]. The routers can act like a computing machine and execute logic functions (transcoding in this work) on the cached data. In this way, each ICN router needs to store only the highest bitrate version for a partial set of video segments and derive other lower bitrate versions based on local online transcoding. Second, for requests for and responses of uncached contents, we design a routing scheme to schedule the traffic load across

the network to reduce video delivery time over ICN. It should be noted that these two challenges are tightly coupled. In general, a higher cache hit ratio needs more cache space and transcoding computation, resulting in higher operational cost for service providers. However, the traffic to the uncached contents can be reduced; therefore, it is profitable to reduce the service latency for those long-tail requests. To balance the trade-off, we propose an optimization framework to orchestrate content management (i.e., caching and transcoding) and the routing scheme.

Our algorithm iteratively solves this problem in two steps. First, given a routing scheme, our system first finds the corresponding optimal configuration of caching and transcoding. Second, based on the previous resource configuration, our system then searches the associated optimal routing scheme. These procedures repeat until convergence.

Our contributions in this article are multiple, including the following.

- By integrating transcoding to the named function, we propose a partial caching scheme to improve the overall cache hit ratio and develop a routing scheme to reduce service latency for uncached contents. Furthermore, we present a set of models for the adaptive video streaming over ICN and formulate an optimization problem for content management and routing control.
- We propose an iterative algorithm to jointly optimize content management, including caching and transcoding, and the routing scheme. In addition, through rigorous mathematical derivations, we prove the convergence and optimality of our proposed approach.
- Through extensive simulations, we find that our strategy achieves significant savings compared with the single optimization scheme, that is, either resource configuration or routing. More importantly, we find that more resources should be allocated to ICN routers with heavier request rates, and the routing scheme favors the shortest path to allocate more traffic.

The rest of this article is organized as follows. Section 2 outlines the related works. Section 3 describes a motivational example and our system architecture, and presents the problem formulation. Section 4 derives the optimal solution. Section 5 presents numerically evaluations. Section 6 summarizes this work.

## 2 RELATED WORKS

In this section, we investigate relevant literature categorized into caching in ICN and video distribution over ICN.

### 2.1 Caching in ICN

One of the salient features in ICN technology is in-network caching, which can improve the efficiency of content distribution. Motivated by the classic ubiquitous in-network caching scheme, Jacobson et al. [21] proposed the Content-centric Network (CCN). Extensive studies had been devoted to exploring cache efficiency from cache size allocation [36], feasibility [19], energy efficiency [26], caching scheme [1], and management [34]. However, the performance gain of ICN was questioned, and several research results were somewhat inconsistent and indecisive. Rossi and Rossini [36] suggested using the degree of centrality of nodes as the indicator for the required cache size and allocating cache size proportionally to the centrality of routers in the network, but the performance gain is very limited. Recent studies [7, 10] show that the performance gain with respect to response time, network congestion, and server load is around 9% to 17% in comparison with the simplest edge caching architecture.

To further improve caching efficiency, some works opened up several new ways, including cache collaboration and coupling caching and routing. Li et al. [30] proposed coordinating a caching

scheme to improve cost efficiency by balancing the trade-off between routing performance and coordination cost. Eum et al. [9] examined the policy of incorporating the original content source and all copies in caches into the routing process. Yeh et al. [44] presented a framework to jointly control network traffic and caching strategies to optimize network performance in view of both current traffic loads and future traffic demands. Rossini and Rossi [37] illustrated the advantages of coupling meta-caching and forwarding. Bilal et al. combined networking coding with ICN to improve caching efficiency [4].

### 2.2 Video Distribution Over ICN

Driven by the exploding growth of global video traffic, some researchers have paid special attention to the study of video distribution over ICN. Westphal et al. [44] surveyed the existing or potential solutions on distributing video streaming over ICN from the engineering aspect and presented a pioneering implementation on integrating adaptive video streaming with ICN [29]. Grandl et al. [13] suggested a pure in-network transcoding scheme to store only the highest-quality segment and derive others by real-time transcoding, aiming to overcome excessive cache use in ICN nodes. Using a countrywide topology and video access traces, Sun et al. [41] evaluated the performance of various content replacement and placement strategies over ICN. Kulinski and Burke [27] implemented NDNVideo as a prototype to distribute streaming video over ICN. Westphal et al. [43] discussed the consequences of the methods to move the underlying network architecture from the current Internet to an ICN architecture on video distribution

Our work is motivated by the recent works on named function of ICN [38, 42]. Named function extends classic ICN, such that, in addition to resolving data access by name, it also supports the concept of function definition and application to data (or other functions) in the same resolution-by-name process. The routers can act like a computing machine and execute logic functions on the cached data. In this work, we enable ICN routers to transcode the cache videos to lower-bitrate versions to improve the cache hit ratio. Our work clearly differs from the aforementioned research as follows. (1) Using the named function [38] of the ICN framework, we introduce the transcoding feature to the ICN router to efficiently deliver adaptive video streaming. (2) We present a partial caching scheme for ICN routers and strategically schedule in-network caching and transcoding to improve overall system performance in terms of cache hit ratio and operational cost from the view of the service provider. (3) By incorporating the routing scheme and caching, we aim to reduce the service delay caused by long-tailed requests to unpopular videos for end users.

## 3 SYSTEM ARCHITECTURE AND PROBLEM FORMULATION

In this section, we begin with a simple example to provide the insight of joint optimization of the cache hit ratio for service providers and service delay for end users. We then give an architectural overview of adaptive video distribution over ICN architecture. Finally, we introduce our system models and formulate an optimization problem.

### 3.1 Motivating Example

For the sake of clarifying the problem of joint optimization of the cache hit ratio for service providers and service delay for end users, we present an illustrative example in Figure 1. The service provider has a central content source and two cache-enabled ICN routers $v_1$ and $v_2$. There are two equal-sized video contents $c_1$ and $c_2$ (10MB) with popularity $c_1 : c_2 = 0.8 : 0.2$ residing in the content source. User requests arrive at the ICN routers $v_1$ and $v_2$ at the rate $\lambda_1 = 5$ and $\lambda_2 = 10$, respectively, which follows the uniform distribution. We assume that the service provider has a limited cache budget to store two copies of contents in ICN routers and provide a sharing access link with a capacity of 50Mbps to $v_1$ and $v_2$ to access the uncached contents. In this article, we

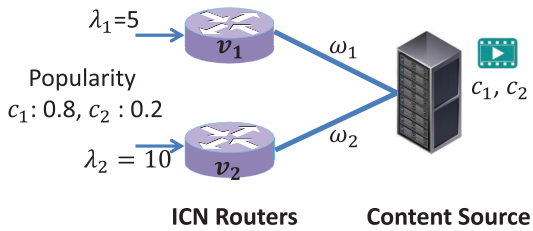



Fig. 1. An example scenario of joint optimization of the cache hit ratio and service delay. End users access edge ICN routers with limited cache capacity for video service. Service providers determine different caching strategies for ICN routers subject to the cache budget and allocate bandwidth resource for them to access uncached video contents.

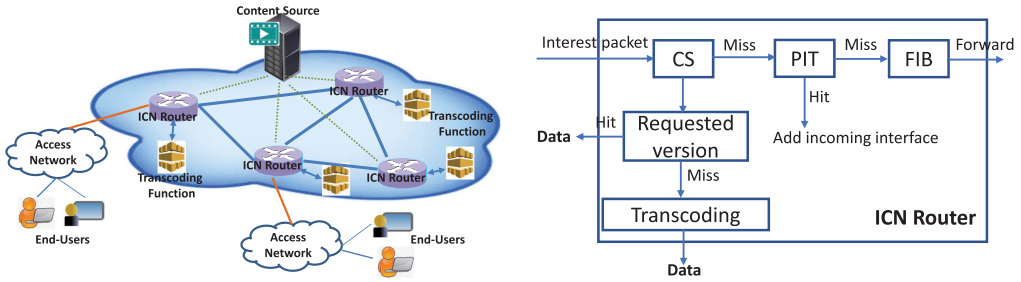
Table 1. Comparison of Averaged Cache Hit Ratio and Service Delay on Different Combinations of Caching Strategy and Routing Scheme in the Motivated Example

| | *Routing1* | *Routing2* | *Routing3* |
|---|---|---|---|
| *Caching1* | hit: 0.33<br>delay: 0.67 | hit: 0.33<br>delay: 0.27 | hit: 0.33<br>delay: 0.18 |
| *Caching2* | **hit: 0.8**<br>delay: 0.15 | **hit: 0.8**<br>**delay: 0.08** | **hit: 0.8**<br>delay: 0.10 |
| *Caching3* | hit: 0.67<br>**delay: 0.08** | hit: 0.67<br>delay: 0.15 | hit: 0.67<br>delay: 0.33 |

assume the existence of a TCP-like mechanism that fairly shares link-bandwidth among $v_1$ and $v_2$ [2]. The links of $v_1$ to content source and $v_2$ to content source have a capacity of $\omega_1$Mbps and $\omega_2$Mbps, respectively. In practice, there may be several intermediate **ICN routers** between one ICN router and the content source. As a result, a path, controlled by the routing policy, will be constructed for this router and content source pair for information exchange. The bandwidth of this path is determined by all the links belonging to this path.

In general, there are three caching schemes to fully use the cache budget, including $v_1$ storing content $c_1$ and $c_2$ (i.e., *Caching1*), $v_1$ and $v_2$ storing content $c_1$, respectively (i.e., *Caching2*), and $v_2$ storing content $c_1$ and $c_2$ (i.e., *Caching3*). We consider three routing schemes to divide bandwidth between $\omega_1$ and $\omega_2$, including $\omega_1 = 40$Mbps, $\omega_2 = 10$Mbps (i.e., *Routing1*), $\omega_1 = 25$Mbps, $\omega_2 = 25$Mbps (i.e., *Routing2*), and $\omega_1 = 10$Mbps, $\omega_2 = 40$Mbps (i.e., *Routing3*). Different combinations of a caching scheme and a routing scheme lead to distinctive averaged cache hit ratios for service providers and averaged service delays for end users. Let us first consider the case of adopting *Caching2* and *Routing3*. In this case, the popular content $c_1$ is stored at $v_1$ and $v_2$; thereby, the averaged cache hit ratio is $(5 \times 0.8 + 10 \times 0.8)/15 = 0.8$. As for the uncached content $c_2$, ICN routers need to access them from the content source. In this work, we omit the access delay from end users to ICN routers and measure the averaged service delay as the video transmission time to clients. The averaged service delay is $(\frac{5\times0.2}{10} + \frac{10\times0.2}{40}) \times 10/15 = 0.1$. Suppose next that we adopt the *Caching*3 and *Routing*1 schemes. The resultant averaged cache hit ratio is 0.67, while the averaged service delay is 0.08. From the view of the service provider, the first combination achieves a better cache hit ratio; for the end users, the second combination is preferred.

Table 1 shows the comparison on all combinations. This table demonstrates that adopting the *Caching2* scheme results in the best averaged cache hit ratio, but the averaged service delay differs




(a) data plane (b) control plane

Fig. 2. System architecture of the proposed framework. (a) *Data plane:* End-to-end view of on-demand adaptive video streaming via ICN backbone network. The end users consume videos in different bitrate versions on diverse devices. Those video segments are originally delivered by the content source to the ICN routers, which strategically adopt in-network cache in local *Content Store* and named function for video transcoding. ICN routers fulfill user requests from the local *Content Store* or the original content source. (b) *Control plane*: The ICN router handles the interest packet according to the cache status, including the exact hit and transcoding hit.

on the different routing schemes. Moreover, the combination of *Caching2* and *Routing2* can simultaneously optimize both metrics. This example motivates us to couple the in-network caching and routing to optimize dual metrics for both service provider and end users.

### 3.2 System Architecture

Figure 2(a) presents a systematic architecture of adaptive video distribution over ICN. The system consists of three parts.

*Content Source*: All videos are divided into a collection of chunks and encoded into several versions with different bitrates. The content source hosts all of those video segments and aims to effectively deliver them to end users.

*ICN Backbone Network*: The content source pushes video segments to the ICN backbone network. Each ICN router runs the ICN name-based publish/subscribe protocol and supports in-network caching and named function [38, 42]. A video access can be further processed by the named function, that is, transcoding, such that requests to the video chunks in the lower-bitrate versions can be served by the higher-bitrate versions via transcoding. Note that we assume that the content provider needs to decide the cache size of different ICN routers to improve the performance network-wide. An ICN router cannot cache all of the segments owing to its limited local cache space, which is usually much smaller than the total volume of contents. All ICN routers participate in the process of content caching in conjunction with their primitive function of relying on the information objects downstream. **For the uncached video segments, ICN routers transmit the interest packet to other ICN routers according to the forwarding policy, commonly toward the content source.**

*End Users*: Geo-diverse end users connect to the corresponding ICN routers via the access network for video service. Due to the diversity of end devices and dynamic network environments, the requested video streaming typically uses various bitrates and formats [16, 23].

In our proposed system architecture, each video has multiple versions and each ICN router can cache only a part of the segments with various versions. When an end user wants to access the required video copy, a retrieval request (*Interest Packet*) is first sent to the nearby ICN router. The packet is processed as illustrated in Figure 2(b) in the following steps. (1) Upon receiving the

interest packet, each router looks up the data name in the local *Content Store* to check whether the requested content is present. If there is an exact copy, the router returns the video segment. If there exists the highest bitrate version, the router executes the named function to transcode it to the requested version and returns back. (2) When the data cannot be served locally, the router checks the *Pending Interest Table* (PIT), which keeps records of unserved interest. If the same record exists, it adds only the incoming interface to this record. Ongoing requests are recorded in a PIT for later sending back the requested data through the reverse path. (3) If there is a PIT miss, the router forwards the interest via the outgoing interface(s) according to its Forwarding Information Base (FIB). Note that PIT records can be used to calculate the access frequency of video segments and assist in determining cache policies. Using this procedure, all of the video requests will be served by either the ICN router or the content source. For video requests served by the content source, our system can configure the routing strategy, including path planning and bandwidth, to determine the service delay.

In this work, we conduct joint optimization for both service providers and end users. From the view of service providers, the key design objective is to increase the cache hit ratio subject to the operational cost, including content caching and transcoding. Specifically, on the one hand, enabling the transcoding feature to ICN routers increases the cache hit ratio, saving the traffic of both the backbone network and content source. On the other hand, certain operational costs will be incurred by using either caching or transcoding resources. From the perspective of end users, service delay plays the vital role in the quality of experience and engagement. Intuitively, service delay is determined by the local cache hit ratio and routing strategy. As such, the optimization metrics for service providers and end users are tightly coupled. In particular, a higher cache ratio leads to more requests and less traffic to the content source. As a result, the service delay will be lessened with the sacrifice of higher operational cost, including storage and computational cost due to transcoding. Conversely, migrating certain video requests that rely on transcoding to the content source may increase the service delay but cut the transcoding cost. Therefore, there is an opportunity to jointly increase the cache hit ratio and lessen the service delay by strategically considering the cache allocation, transcoding configuration, and routing scheme.

### 3.3 System Model

In this section, we present the system model in four components to capture key features of the system. For clarity of discussion, we summarize the important notations in Table 2.

*3.3.1 Network Topology.* We model the video distribution network as an undirected graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_N\}$ is the set of ICN routers and $E$ denotes network links between those routers. For simplicity, we consider that there is only one content source $v_0$. Note that this network model can be easily extended to multiple content sources by connecting them with a direct link with infinite capacity to a "single super server."

Each node $v_n$ is associated with content caching and transcoding. The consumption on each resource will incur a corresponding amount of cost in a pay-per-use manner. Each edge $e_{nn'}$ represents the link between node $v_n$ and $v_{n'}$, which has limited bandwidth capacity $d_{e_{nn'}}$. Content source $v_0$ distributes video segments to other nodes via edges subject to the link capacity. When delivering video segments to node $v_n$ of interest, there exists multiple transmission paths with different hop counts. The content source can make traffic engineering for path planning.

*3.3.2 Content Model.* Assume that there are $I$ different video segments in the content source to be distributed, where each segment has $J$ different bitrate versions (without loss of generality, $J > 1$), $b_1, b_2, \ldots, b_J$, ranging from the highest bitrate $b_1$ to the lowest bitrate $b_J$. We assume that the length of each segment is the same, on average. Thereby, different video segments with the

Table 2. Notations

| Symbol | Descriptions |
|---|---|
| $G = (V, E)$ | $V$: router set, E: link set |
| $v_1, v_2, \ldots, v_n$ | ICN routers |
| $e_{nn'}$, $d_{e_{nn'}}$ | Link between router $v_n$ and $v_{n'}$ and the corresponding bandwidth capacity |
| $i = 1, 2, \ldots, I$ | Video popularity rank index from 1 to maximum $I$ |
| $j = 1, 2, \ldots, J$ | Bitrate version index from 1 to maximum $J$ |
| $b_1, b_2, \ldots, b_J$ | Bitrates of different versions |
| $s_{ij}$ | The $i$th video segment in the $j$th bitrate version |
| $P(s_{ij}), p_j$ | Video access probability to segment $s_{ij}$ and the $j$th bitrate version |
| $z_n, x_n, y_n$ | Cache capacity of router $v_n$, cached all the bitrate versions for the top popular videos, and highest bitrate version for other videos |
| $P_n^{Hit}, P_n^{Miss}, P_n^{Tr}$ | Local cache hit ratio, local miss ratio, and local transcoding ratio |
| $(R_{nk}, k = 1, 2, \ldots, K)$ | Routing paths to content source for router $v_n$ |
| $\pi_n^k$ | Traffic ratio allocated to path $k$ |
| $O_e$ | Traffic volume passed by link $e$ |
| $\mathfrak{R}_{hit}, \mathfrak{R}_{cache}, \mathfrak{R}_{trans}$ | The overall cache hit ratio, cache cost, and transcoding cost |
| $\mathcal{T}$ | Overall service latency |

same bitrate have an equal size. All of those $J$ bitrates are frequently accessed (i.e., we do not consider the versions that are seldom used). As such, we have overall $I * J$ different versions of all video segments. Each segment is denoted as $s_{ij}$, where $i = 1, 2, \ldots, I$ refers to its popularity rank, and $j = 1, 2, \ldots, J$ refers to its bitrate rank.

According to the observation in [45], video segment popularities follow a Zipf distribution, like the distribution of web objects' popularities on the Internet. We assume that the aggregated user requests toward those $I$ different segments follow a Zipf-like distribution. In addition, we adopt the assumption from [14, 30, 47] that video requests arrive at each ICN router independently (i.e., independent reference model) and that their arrival patterns all follow the same Poisson process with the arrival rate as $\lambda_n$ at ICN router $v_n$. Thus, the probability of requesting the $j$th bitrate version of the $i$th popular segment $s_{ij}$ at a router is

$$P(s_{ij}) = \frac{p_j \times 1/i^{\alpha}}{\sum_{k=1}^{I}(1/k^{\alpha})} = \frac{p_j \times 1/i^{\alpha}}{H_{I,\alpha}}, i = 1, \ldots, I, \tag{1}$$

where $p_j$ is the probability of requesting the $j$th bit rate of a segment, $H_{I,\alpha} = \sum_{k=1}^{I}(1/k^{\alpha})$ is the $I$th generalized harmonic number, and $\alpha$ is the shape parameter of the Zipf distribution ($\alpha$ must be positive). Although we assume that $p_j$ is independent of the router $v_n$ and access probability of the $i$th video segment, our model can be easily extended to more generic cases by modifying Equation (5). A large $\alpha$ indicates the relatively small percentage of very popular ones. Typically, $\alpha$ is between 0.5 and 1.5, (e.g., [10] found that $\alpha$ is around 1).

*3.3.3 Partial Caching and Request Model.* We adopt the partial in-network caching and transcoding model as shown in Figure 3. Following the descending order of video segment popularity rank, each ICN router $v_n$ stores all of the bitrate versions for the top $x_n$ popular video segments and the highest bitrate version for other less popular segments, constrained by the allocated cache space $z_n$. In total, the top $y_n$ popular video segments will be cached, of which we store the highest bitrate versions only for the video segments whose popularities fall in the range $[x_n + 1, y_n]$, as

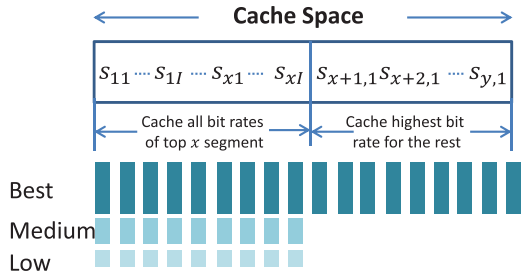



Fig. 3. Illustration of the partial caching scheme. For each ICN router, we store all the bitrate versions only for the top $x$ popular videos and the highest bitrate version for the rest, subject to the cache space.

illustrated in Figure 3. We have the following equation:

$$z_n = x_n \sum_{j=1}^{J} b_j + (y_n - x_n)b_1 = x_n B_s + (y_n - x_n)b_1, \tag{2}$$

where $B_s = \sum_{j=1}^{J} b_j$ is the total size of all bitrate versions of one video segment. Let $\bar{b} = \sum_{j=1}^{J} p_j b_j$ denote the average bitrate in regard to the probability of different bitrate versions. We assume that the cache space $z_n$ is too small to host all of the video segments, and the transcoding resource is capable for the less popular video segments $y_n - x_n$. From the equation, we have that $x_n \in [0, z_n/B_s]$.

The partial caching scheme is a generalization of two extreme cases, that is, *all rate caching* [21] and *pure transcoding* [13]. For the all-rate caching scheme, different bitrate versions of the same video segment are treated as different segments. All of them are cached by all of the ICN routers. For the pure transcoding scheme, it keeps only the highest bitrate version for all videos. Once receiving a request for a lower bitrate version, it will transcode the highest bitrate version to the desired one. As illustrated in Figure 3, our scheme will degrade to the all-rate caching scheme when $x = y$ and to the pure transcoding scheme when $x = 0$.

Under this partial caching scheme, for each ICN router $v_n$, there will be three possible states to serve user requests:

- *Exact Hit*: The requested video segment with the required bitrate version is residing in the local store. In this case, this copy is returned back without other expenses on computing.
- *Transcoding Hit*: The requested video segment is residing in the local store but with a different bitrate version, that is, having only the highest bitrate version. This request can be served by transcoding the highest bitrate version into the desired one, incurring computing cost.
- *Cache Miss*: The requested video segment is not in the local store. In this case, we assume that all cache misses will be directly forwarded to the content source.

Therefore, the local cache hit ratio $P_n^{Hit}$, including both the exact hit and transcoding hit, can be derived as

$$P_n^{Hit} = \sum_{i=1}^{y_n} \sum_{j=1}^{J} P(s_{ij}) = \frac{H_{y_n,\alpha}}{H_{I,\alpha}}, \tag{3}$$

and the local miss ratio $P_n^{Miss}$ is

$$P_n^{Miss} = 1 - P_n^{Hit}. \tag{4}$$

The probability of local transcoding ratio $P_n^{Tr}$ is calculated as

$$P_n^{Tr} = \sum_{i=x_n+1}^{y_n} \sum_{j=2}^{J} P(s_{ij}) = \frac{(1-p_1)\left(H_{y_n,\alpha} - H_{x_n,\alpha}\right)}{H_{I,\alpha}}. \tag{5}$$

In order to ease the analysis and derive meaningful results, we assume that the total segment amount $I$ is sufficiently large ($I >> 1$) and the Zipf parameter $\alpha$ cannot be exactly equal to 1 (but can be arbitrarily close to 1). Thus, we approximate the local cache hit ratio by using a continuous function as

$$P_n^{Hit} \approx \frac{\int_1^{y_n} t^{-\alpha} dt}{\int_1^{I} t^{-\alpha} dt} = \frac{y_n^{1-\alpha} - 1}{I^{1-\alpha} - 1},\ \alpha > 0,\ \alpha \neq 1.$$

Similarly, we approximate the local transcoding ratio as

$$P_n^{Tr} \approx \frac{y_n^{1-\alpha} - x_n^{1-\alpha}}{I^{1-\alpha} - 1}(1-p_1),\ \alpha > 0,\ \alpha \neq 1.$$

*3.3.4 Routing Model.* When caching misses occur, those video requests will be served by the content source. In other words, the links to the content source may become the bottleneck, which further affects the service delay of end users. In this work, our system strategically plans the routing scheme at the content source to alleviate the drawback. We assume that the content source has a global view, in terms of congestion level along each path, of the whole ICN backbone network. Upon the arrival of video requests, the content source performs the allocation based on the awareness of congestion level over the paths that the flows can take in the network.

For each pair of content source $v_0$ and ICN router $v_n$, we have a set of $K_n$ distinct paths ($R_{nk}, k = 1, 2, \ldots, K_n$) for content delivery, where a path is an acyclic sequence of links $e \in E$ going from $v_0$ to $v_n$. If a link $e$ belongs to path $R_{nk}$, we say that $e \in R_{nk}$. The set of $\{K_n, n = 1, 2, \ldots, N\}$ forms the traffic matrix of the ICN backbone network. In general, the traffic matrix is relatively large to guarantee the path diversity and robust for content distribution. We assume that the amount of traffic to router $v_n$ can be allocated to the set of paths $R_{nk}$. Let coefficient $\pi_n^k$ ($\pi_n^k \geq 0$) denote the traffic ratio allocated to the path $k$, which satisfies

$$\sum_{k=1}^{K_n} \pi_n^k = 1, \qquad \forall n. \tag{6}$$

When $v_0$ delivers video segments to a set of nodes, a link $e$ may be shared by multiple flows. We assume that a transport mechanism (some forms of interest-shaping in ICN) shares the bandwidth [40]. For any link $e$, the traffic volume that passes by this link can be calculated as

$$O_e = \sum_{n=1}^{N} \sum_{k=1}^{K_n} \lambda_n \bar{b} P_n^{Miss} \pi_n^k \mathbb{I}_{e,R_{nk}}, \qquad \forall e, \tag{7}$$

where $\lambda_n \bar{b} P_n^{Miss}$ is the traffic volume to router $v_n$ and $\mathbb{I}_{e,R_k}$ is an indicator function. $\mathbb{I}_{e,R_{nk}} = 1$ only when $e \in R_{nk}$. In general, the system needs to guarantee that the traffic volume through a link is less than the link capacity, that is, $O_e \leq d_e$. When $O_e > d_e$, this link incurs congestion, resulting in longer service latency. To simplify the analysis, we assume that $O_e \leq d_e, \forall e$. For any link $e$, given the traffic volume through this link and the link capacity, we can calculate the service delay passed by this link, denoted as $\tau(O_e, d_e)$. We assume that $\tau(O_e, d_e)$ is strictly increasing and convex. In practice, when there is no congestion, we can use the hop count as the service delay function. Other complicated forms, such as an exponential function or a piecewise linear function, can be constructed for different environments, such as congestion.

### 3.4 Problem Formulation

In this section, we first formulate the optimization problems for service providers and end users separately by considering the content management, including caching and transcoding, and the routing scheme. Then, we create a joint optimization for both parties.

*3.4.1 Optimization for Service Provider.* From the perspective of service providers, the design objective is to maximize the overall cache hit ratio for a given routing matrix. As such, more requests can be served at ICN routers, and the traffic to the content source is reduced in the meantime. Using the system models, the overall cache hit ratio is given by

$$\mathfrak{R}_{hit} = \sum_{n=1}^{N} P_n^{Hit} \lambda_n. \tag{8}$$

The cache cost is defined as

$$\mathfrak{R}_{cache} = \sum_{n=1}^{N} z_n. \tag{9}$$

Transcoding cost can be in the monetary domain, temporal domain, spatial domain, and the quantization domain, and the resulting computational complexity can be quite different. Zhang et al. [46] modeled the relationship between the video transcoding time and file size using a linear function $A = LX$ via extensive experiments, where $A$ is the transcoding time, $L$ is the file size, and $X$ is a random parameter following gamma distribution. The predominant cloud service providers, including Windows Azure and Amazon, price the transcoding cost into a linear function. In this work, we adopt a similar linear model and express the transcoding cost as

$$\mathfrak{R}_{trans} = \sum_{n=1}^{N} \lambda_n P_n^{Tr} \sum_{j=2}^{J} p_j b_j = \sum_{n=1}^{N} \lambda_n P_n^{Tr} (\overline{b} - p_1 b_1), \tag{10}$$

where $\overline{b}$ denotes the average size of a video segment in different bitrate versions. Therefore, we have the following optimization problem:

$$\mathbf{P1}: \max_{x_n, y_n} \quad \mathfrak{R}_{hit}, \tag{11}$$

$$s.t. \quad \mathfrak{R}_{cache} \leq \beta_1, \tag{12}$$

$$\mathfrak{R}_{trans} \leq \beta_2, \tag{13}$$

$$O_e \leq d_e, \quad \forall e, \tag{14}$$

where the decision variable in the objective function (11) is the caching allocation $x_n$ and $y_n$, the constraint (12) captures the limitation of the given cache budget $\beta_1$ on the overall caching space, the constraint (13) guarantees that the overall transcoding cost is less than the given limitation $\beta_2$, and constraint (14) refers to the link limitation.

*3.4.2 Optimization for End Users.* From the view of end users, service delay affects users' quality of experience and engagement. It comprises two parts, service delay from end users to ICN routers (over the access network) and service delay from the ICN routers to the content source. Considering the current progress of wired/wireless broadband technology, the service delay on the access network is much smaller than that on the ICN backbone network. To simplify the analysis, we consider only the service delay on the ICN backbone network caused by the delivery of uncached video segments. However, our framework can be easily extended to more complicated

cases. The service latency $\mathcal{T}$ is given by

$$\mathcal{T} = \sum_{e \in E} \tau(O_e, d_e). \tag{15}$$

Given the configuration $(x_n, y_n)$ of transcoding and caching resources, we model this problem as follows:

$$\begin{aligned} \mathbf{P2}: \min_{\pi_n^k} \quad & \mathcal{T}, \\ s.t. \quad & \text{constraint (6)}. \end{aligned} \tag{16}$$

*3.4.3 Joint Optimization.* By jointly considering the resource configuration of content caching and transcoding, and the routing scheme, the objective is to simultaneously maximize the cache hit ratio and minimize the service delay, which is formally given by

$$\begin{aligned} \mathbf{P3}: \ & \{x_n^*, y_n^*, \pi_n^{k^*}\} = \arg\{\min_{\pi_n^k} \mathcal{T}, \max_{x_n, y_n} \mathfrak{R}_{hit}\}, \\ & s.t. \quad \text{constraint (12), (13), (14) and (6)}. \end{aligned} \tag{17}$$

## 4 ALGORITHMS AND ANALYSIS

In this section, we first describe an iterative approach to solving the problem. We then rigorously prove the convergence of the proposed approach. Finally, we present two algorithms to find the optimal solution.

### 4.1 Iterative Approach

The optimization objective function (Equation (17)) is a combination of two objective functions, including cache hit ratio (Equation (11)) and service delay (Equation (16)). However, these two metrics are coupled. On the one hand, the optimization of the cache hit ratio is constrained by link capacity, which is determined by the control variable $\pi_n^k$ of the service delay function. On the other hand, the control variables $x_n$ and $y_n$ of the cache hit ratio function decide the cache missing ratio $P_n^{Miss}$, which affects the optimization of the service delay function.

In this work, we propose an iterative approach to find the optimal solution. First, given an initial resource allocation policy, we solve the optimization problem **P2** to find a routing scheme $\pi_n^k, \forall 1 \le n \le N, 1 \le k \le K_n$. Then, we apply this routing scheme to the optimization problem **P1** to find a new resource allocation policy. These iterations repeat again and again until the resource allocation variable $x_n$ and $y_n$ converge. Detailed procedures are illustrated in Algorithm 1.

**ALGORITHM 1:** Iterative Approach to Solving Problem **P3**

**Input**: Initial resource allocation $\{x_n[0], y_n[0], \forall 1 \le n \le N\}$, Predefined thresholds $\epsilon_1$ and $\epsilon_2$ for iteration termination.

**Output**: Optimal resource allocation and routing policies $\{x_n^*, y_n^*, \pi_n^{k^*}, \forall 1 \le n \le N, \forall 1 \le k \le K_n\}$.

$m = 1$;

**do**

Solve the optimization problem **P2** to find a routing scheme $\{\pi_n^k[m], \forall 1 \le n \le N, \forall 1 \le k \le K_n\}$;

Update $m = m + 1$;

Solve the optimization problem **P1** to find a resource allocation policy $\{x_n[m], y_n[m], \forall 1 \le n \le N\}$;

**while** $|x_n[m+1] - x_n[m]| > \epsilon_1$ *or* $|y_n[m+1] - y_n[m]| > \epsilon_2, \forall n$;

$x_n^* = x_n[m], y_n^* = y_n[m], \pi_n^{k^*} = \pi_n^k[m], \forall 1 \le n \le N, \forall 1 \le k \le K_n$;

### 4.2 Convergence

Using Algorithm 1, we need to solve problems **P1** and **P2** and guarantee the convergence of the iterations. In this section, we first prove that we can find the optimums of problems **P1** and **P2** using their convex property. Second, based on the Gauss-Seidel method [3], we construct an equivalent problem to **P3** and prove its convergence to the optimality.

*4.2.1 Convergence of Problems **P1** and **P2**.* By checking the convexity of Equation (11) to Equation (14) with respect to the control variables, we can prove that problem **P1** is a convex optimization problem. Specifically, we derive their second-order derivatives and check the positivity, as described in the following.

Lemma 1. *All of the equations in problem **P1** are convex with respect to $x_n$ and $u_n$, where $u_n = y_n^{1-\alpha}$. Problem **P1** can be transformed into a convex optimization problem.*

Proof. See Appendix A.1 for a complete proof. □

Lemma 2. *All of the equations in problem **P2** are convex with respect to $\pi_n^k$. Problem **P2** is a convex optimization problem.*

Proof. We assume that $\tau(O_e, d_e)$ is strictly convex in $\pi_n^k$, and the constraint (6) is a linear function of $\pi_n^k$. Therefore, problem **P2** is a convex optimization problem in $\pi_n^k$. □

Based on Lemmas 1 and 2, we can find the optimums of problems **P1** and **P2**.

*4.2.2 Convergence of Problem **P3**.* To solve problem **P3**, we consider a special case of **P3**, that is, the appropriate weighted average, and construct an objective function by combining two objective functions (i.e., Equations (11) and (16)) as follows:

$$\overrightarrow{C}(x_n, u_n, \pi_n^k) = \mathcal{T} - \mathfrak{R}_{hit}. \tag{18}$$

In general, we need a weight factor to balance two optimization objectives $\mathcal{T}$ and $\mathfrak{R}_{hit}$. To ease our analysis, we assume that these two optimization objectives are in the same unit. However, our analysis can be easily extended to the generic case. The original problem **P1** is to maximize the cache hit ratio $\mathfrak{R}_{hit}$; problem **P2** is to minimize the delay $\mathcal{T}$. The objectives of problems **P1** and **P2** are equivalent to minimize the objective function (18), making the routing scheme $\pi_n^k$ correspond exactly with the optimal resource allocations $x_n$ and $u_n$. Thus, the convergence of problem **P3** is equivalent to prove the convergence of the following problem:

$$\mathbf{P4}: \min_{x_n, u_n, \pi_n^k} \quad \overrightarrow{C}(x_n, u_n, \pi_n^k), \tag{19}$$
$$s.t. \quad \text{constraints (12), (13), (14), and (6).}$$

Based on Lemmas 1 and 2, we complete the convergence proof in the following theorem.

Theorem 1. *The joint optimization problem **P4** converges to its optimum.*

Proof. See Appendix A.2 for a complete proof. □

### 4.3 Algorithms

We propose two algorithms to solve problems **P2** and **P1** in Steps 3 and 5 of Algorithm 1 separately. In particular, since the control variable $\pi_n^k$ in problem **P2** is continuous, we adopt the Karush-Kuhn-Tucker (KKT) conditions to find the optimal routing scheme for problem **P2**. Moreover, we adopt a subgradient method to find the optimal content management for problem **P1**.

*4.3.1 Algorithm to Solve Problem **P2**.* Given a resource configuration $\{x_n, y_n, \forall 1 \le n \le N\}$, we use KKT conditions to solve the constrained problem. By introducing a Lagrange multiplier $\gamma$ to constraint (6), the Lagrange function is defined as

$$L\left(\pi_n^k, \gamma\right) = \sum_{e \in E} \tau(O_e, d_e) + \gamma \sum_{n=1}^{N} \left( \sum_{k=1}^{K_n} \pi_n^k - 1 \right). \tag{20}$$

Thus, setting the gradient $\nabla L(\pi_n^k, \gamma) = 0$ yields the following equations for the optimal solution,

$$\frac{\partial L(\pi_n^k, \gamma)}{\partial R_{lv}} = \sum_{e \in E} \frac{\partial \tau(O_e, d_e)}{\partial \pi_n^k} + \gamma = 0, \tag{21}$$

$$\frac{\partial L(\pi_n^k, \gamma)}{\partial \gamma} = \sum_{k=1}^{K_n} \pi_n^k - 1 = 0, \forall n = 1, 2, \ldots, N, \tag{22}$$

$$\pi_n^k \mu_n^k = 0, \forall n = 1, 2, \ldots, N, \forall k = 1, 2, \ldots, K_n, \tag{23}$$

where Equation (23) is the complementary slackness condition with $\mu_n^k$ as the KKT multipliers, capturing the positive constraint of $\pi_n^k$.

*4.3.2 Algorithm to Solve Problem **P1**.* Given a routing scheme $\{\pi_n^k, \forall 1 \le n \le N, 1 \le k \le K_n\}$, we use a subgradient approach to find the optimal resource allocation policy, as shown in Algorithm 2. In particular, the algorithm begins with an initial feasible solution $\{x_n[0], u_n[0], \forall 1 \le n \le N\}$. At each iteration, it takes a step $\sigma_m$ along with the subgradient of the objective $\mathfrak{R}_{hit}^{(k)}$ or one of the constraint functions ($\mathfrak{R}_{cache}$, $\mathfrak{R}_{trans}$, or $O_e$). The optimal criterion of this process is that, if the current point is feasible, it uses an objective subgradient; otherwise, the algorithm chooses a subgradient of any violated constraint. We repeat the iteration until it converges. The proof on the convergence and optimality of applying a subgradient method to solve the constrained convex problem can be found in [3].

**ALGORITHM 2:** Subgradient Algorithm to Solve Problem **P1**

**Input**: Routing scheme $\{\pi_n^k, \forall 1 \le n \le N, \forall 1 \le k \le K_n\}$, initial resource allocation $\{x_n[0], u_n[0], \forall 1 \le n \le N\}$, predefined threshold $\epsilon_3$ for iteration termination.
**Output**: Optimal resource allocation $\{x_n^*, u_n^*, \forall 1 \le n \le N\}$.

```
m = 1;
do
    if Constraints (12)(13)(14) are all met. then
        g[m] = ∇ℜ_hit;
    else if Constraint (12) is violated. then
        g[m] = ∇ℜ_cache;
    else if Constraint (13) is violated. then
        g[m] = ∇ℜ_trans;
    else if Constraint (14) is violated. then
        g[m] = ∇O_e;
    update x_n[m + 1] = x_n[m] − σ_m g[m];
    update u_n[m + 1] = u_n[m] − σ_m g[m];
    update m = m + 1;
    update σ_m = 1/m;
while |u_n[m + 1] − u_n[m]| > ε_3, ∃n or any of constraints (12)(13)(14) is violated;
x_n^*[m] = x_n[m] and u_n^*[m] = u_n[m], ∀1 ≤ n ≤ N;
```

Table 3. Bitrate Versions in a Real Adaptive Streaming System

| **Type** | 240p | 360p | 480p | 720p | 1080p |
|---|---|---|---|---|---|
| **Bitrate** | 0.2Mbps | 0.5Mbps | 1.2Mbps | 2.0Mbps | 3.0Mbps |

### 4.4 Complexity Analysis

The complexity of Algorithm 1 relies on two subproblems **P1** and **P2**. For problem **P1**, there are $2N|K| + 1$ equations with $2N|K| + 1$ unknown variables. We can directly solve these equations to obtain the optimal solution. For problem **P2**, it contains $2N$ variables and $|E| + 2$ constraints. The complexity of Algorithm 2 is $O\left((2N + |E|)/\epsilon_3^2\right)$, and it will take $O(1/\epsilon_3^2)$ iterations to converge [3].

## 5 PERFORMANCE EVALUATION

In this section, based on realistic settings, we evaluate the performance of our proposed approach via extensive simulations. Simulation results verify the convergence of the iterative approach, the operational guidelines for resource configuration and routing, and the performance gain over two baseline algorithms.

### 5.1 Parameter Settings

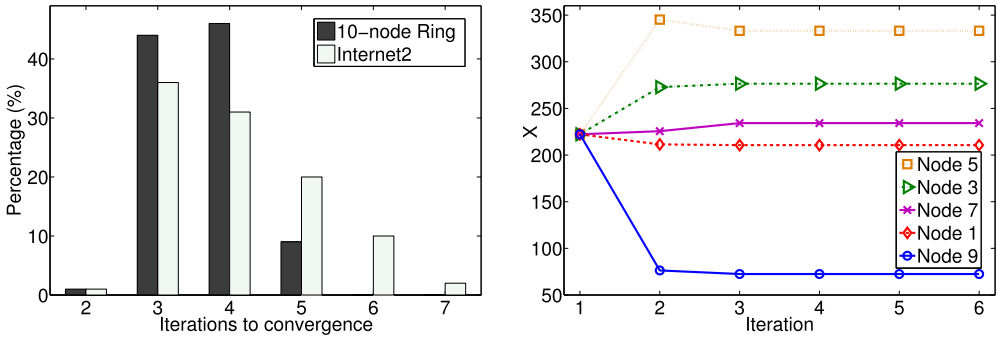
The network topologies used in our simulation include a simple ring with 10 nodes and a real Internet2's layer 3 network in the United States [20], which was built for research and testing purposes for the future Internet. All of the nodes and links are numbered for ease of discussion. The path diversity of two topologies is different: for any two nodes in the ring topology, there are only two connection paths (i.e., clockwise and counter-clockwise); while for that in Internet2, we limit the number of connection paths to $K_n = 4$, which have less hops than other possible paths. In addition, we assume that the link capacity follows a truncated Gaussian distribution bounded below by zero [35], where $d_e \sim \mathcal{N}(1, 0.15)$ Gbps. The network delay function is defined as $\tau(O_e, d_e) = \kappa(\frac{O_e}{d_e})^\kappa$ with order $\kappa = 3$ [35], which implies that the service delay depends on link use.

For the content model, we assume that there are in total $I = 20{,}000$ different video segments with equal duration (one second), each of which has $J = 5$ different bitrate versions. The bitrate setting is from a real adaptive media streaming system [28], as shown in Table 3. The popularity of different video segments follows Zipf distribution with shaping parameter $\alpha = 0.9$; the popularity of different versions follows a truncated Gaussian distribution as $p_j \sim \mathcal{N}(0.2, 0.02)$. We adopt the independent reference model for the arrival rate $\lambda_n$ at each router, which follows a truncated Gaussian distribution [35] as $\lambda_n \sim \mathcal{N}(200{,}20)$ requests per second. In addition, we assume that the total caching resource (**i.e., in file size**) is 10% of all segments, that is, $\beta_1 = 10{,}000$, and the total transcoding resource can transcode 1% segments in real time, that is, $\beta_2 = 1{,}000 \times (\overline{b} - p_1 b_1)$.

### 5.2 Convergence Verification

In this section, we verify the convergence of our proposed Algorithm 1 as stated in Theorem 1. We run the algorithm 100 times with random configuration and obtain the convergence rate in terms of iteration count. In each simulation running, for both topologies, we randomly select a node as the content source and initialize all routers with the same configuration, that is, uniformly assign cache size ($x_n$ and $y_n$) to all routers. All condition thresholds ($\epsilon_1$, $\epsilon_2$, and $\epsilon_3$) used to control the convergence are set as 0.001.

Figure 4(a) shows the iteration count distribution ($m$ in Algorithm 1) for both topologies. We can observe that our proposed algorithm converges to the optimum with 7 iterations for all 100




(a) Convergence rate (b) Cache size allocation for selected routers at Internet2

Fig. 4. Convergence performance of our iterative approach: (a) shows the convergence rate of two topologies; (b) shows the dynamics of cache size allocation for selected routers. The results indicate that our proposed approach converges to the optimum within 7 iterations, and the convergence rate of the 10-node ring is faster than that of the Internet2 topology.

simulations, and more than 65% of cases get converged within 4 iterations. We also find that the convergence rate of the 10-node ring is faster than that of the Internet2 topology. The reason is that, when applying the Gauss-Seidel type iteration approach, the solution at each iteration follows conjugate directions of its decision variables. For each pair of router and content source, the possible routing path $K_n$ is 2; it is 4 for the Internet2 layer3 topology. The number of control variables on routing for the 10-node ring is much less than that of the Internet2 topology. This leads to the faster convergence for the ring topology.

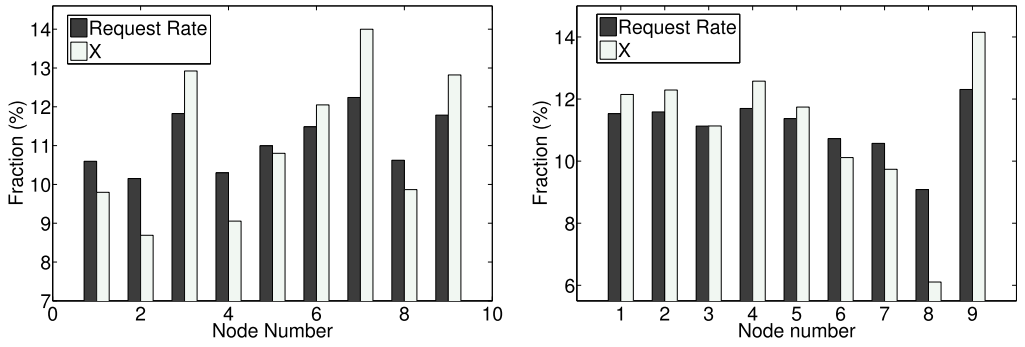
As an example, we plot the cache size $z_n$ evolution for several selected nodes in Internet2 in a simulation (the content source is node 10), as shown in Figure 4(b). We can observe that, after only 3 iterations, the cache size of all selected nodes converges to their optimums.

### 5.3 Joint Optimal Solution

In this section, we investigate the structure of the optimal resource allocation and routing strategy to find meaningful design guidelines. In both topologies, we set node 10 as the content source.

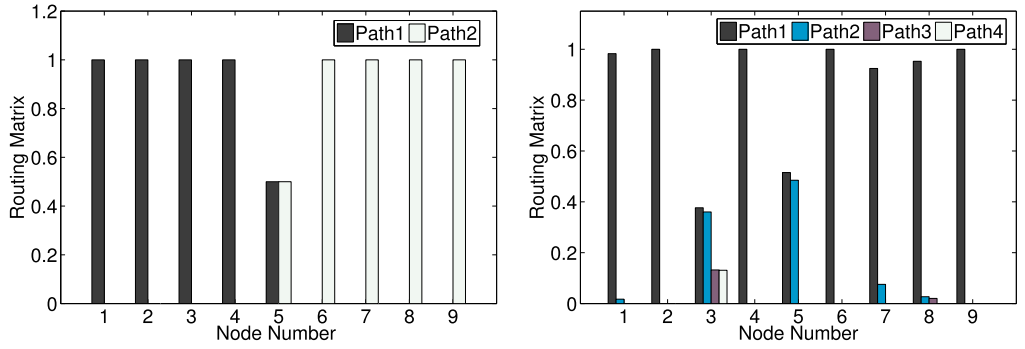
*5.3.1 Optimal Resource Allocation Scheme.* Figure 5 illustrates the correlation of optimal cache size allocation and the corresponding request arrival rate at each node for both topologies, where the y axis denotes the fraction of the allocated cache resource or request rate at each node (i.e., $\lambda_n / \sum_{n=1}^{N} \lambda_n$, and $y_n / \sum_{n}^{N} y_n$). The results reveal that cache size allocation is correlated with the corresponding request arrival rate. More cache size should be allocated to the node with the larger request arrival rate. In Figure 5(a), node 2 has the lowest request rate and the lowest resource amount. In contrast, node 7 has the highest request rate and the highest resource amount. The same phenomenon can be found in Figure 5(b). The reason is the load balance nature of our optimization problem. In particular, on the one hand, given the identical popularity distribution at each node, more requests lead to more local cache misses. On the other hand, by placing more resources, we can reduce those cache misses. As a result, to balance the cache miss traffic at each node, we need to coordinate the resource allocation with the request arrival rate.

*5.3.2 Optimal Network Routing Policy.* For each pair of content source and router, there are several paths (2 in the 10-node ring, and 4 in Internet2) for routing, that is, assigning a weight $\pi_n^k$ on each path for traffic division. Figure 6 illustrates the optimal routing scheme for different nodes in both topologies. The results reveal that the traffic volume assigned to each path is correlated to




(a) Optimal cache allocation over ring (b) Optimal cache allocation over Internet2

Fig. 5. Optimal cache allocation: (a) shows the cache allocation and the corresponding request arrival rate for the 10-node ring; (b) shows the cache allocation and the corresponding request arrival rate for Internet2. The results reveal that cache size allocation is correlated with the corresponding request arrival rate. More cache size should be allocated to the node with the larger request arrival rate.



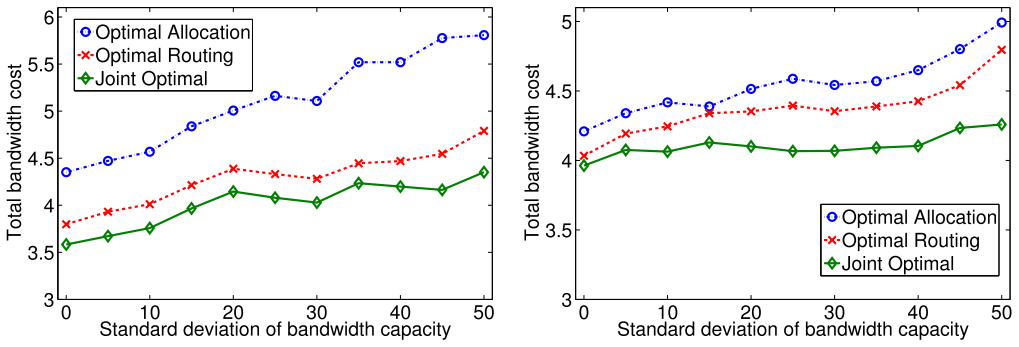

(a) Optimal routing over 10-node ring (b) Optimal routing over Internet2

Fig. 6. Optimal network routing policy: (a) shows the routing policy for the 10-node ring; (b) shows the routing policy for Internet2. The results reveal that the traffic volume assigned to each path is correlated to its hop distance to the content source. The path with less hop distance to the content source will assign more traffic, and paths with the same hop distance will have identical traffic.

its hop distance to the content source. The path with less hop distance to the content source will assign more traffic, and paths with the same hop distance will have identical traffic. In Figure 6(a), all nodes except node 5 have only one shortest path, and they use it only to forward traffic for the uncached contents. Node 5 splits its traffic evenly across two paths, both of which are 5 hops away from the content source. In Figure 6(b), nodes 4, 6, and 9 use only their 1-hop path, whereas node 3 splits the traffic over 4 different paths, which are all 3 hops away from the content source. The reason is that using the shortest path to transmit data minimizes the traversed hops as well as service delay, and splitting the traffic across multiple shortest paths balances the traffic load at each link.

### 5.4 Performance Comparison

Since there are quite a few efforts on video over ICN, we compare the performance of our proposed approach against two baseline algorithms with moderate modifications from relevant works, which optimize only the resource allocation or routing:




(a) Service delay in 10-node ring

(b) Service delay in Internet2

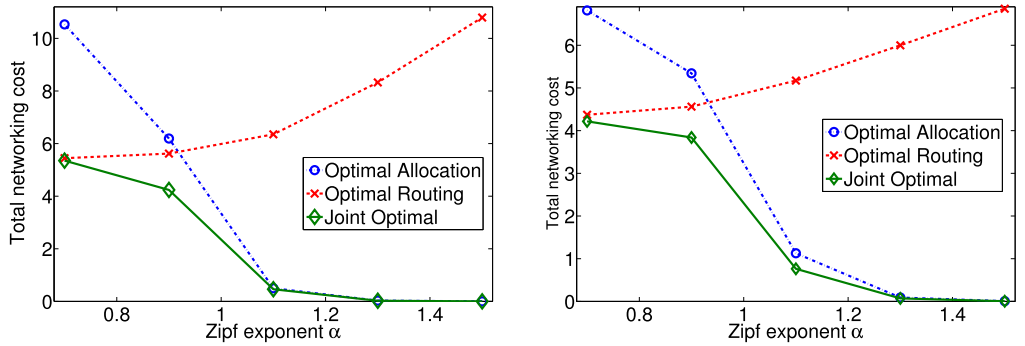
Fig. 7. Service delay comparison of our proposed solution and alternative algorithms under different deviations of arrival rate: (a) shows the comparison results on the 10-node ring topology; (b) shows the comparison results on Internet2 topology. Our proposed algorithm outperforms the other two solutions, and the delay savings will become larger with the increase of deviation.

- *Optimal Allocation:* This scheme is from [34]. The objective is to optimize the cache hit ratio under the configuration that every node always retrieves cache misses by using only one path with the shortest hop distance to the content source.
- *Optimal Routing:* This scheme is from [30], and we adopt the common caching policy, that is, when routers cache a video segment, it means that all of the bitrate versions will be stored. The objective is to optimize the service delay under the configuration that content caching and transcoding resources are always uniformly distributed over all nodes.

We evaluate their performance with various system parameters to obtain some operational guidelines. Similarly, we run 10 random configurations for each setting and report only the average value.

*5.4.1 Service Delay Under Different Deviations of Arrival Rate $\lambda_n$.* The deviation of request arrival rate determines the request distribution over the topology. Larger deviation results in some nodes possibly having more requests while others may have quite a few requests. This imbalance is consistent with the service pattern in the practical geo-distributed networking architecture. We vary the deviation of $\lambda_n$ from 0 to 50.

Figure 7 shows the service delay of three algorithms. The results reveal several insights. First, we find that our joint optimization solution outperforms other baselines in all simulation cases, including both Figure 7(a) and Figure 7(b). This verifies the optimality of our method. In the 10-node ring network topology, our proposed algorithm can reduce the delay from 5.9% to 14.6% and from 1.9% to 11.4% over the optimal allocation strategy and optimal routing strategy, respectively; in the Internet 2 network topology, our proposed algorithm can reduce the delay from 21.8% to 29.4% and from 1.7% to 5.4% over the optimal allocation strategy and optimal routing strategy, respectively. Second, the delay saving in the ring topology (see Figure 7(a)) is higher than the one in Internet2 (see Figure 7(b)). This is because the path diversity of Internet2 is higher than that of ring topology, the congestion penalty can be successfully alleviated via multiple paths. Third, as the increase of deviation, the delay saving of our algorithm becomes larger. The reason is that our joint optimization scheme can make multiple decisions (resource allocation and routing) to tackle the challenges of request imbalance. Finally, the optimal routing scheme outperforms the optimal allocation scheme. This can be easily understood from the optimization objective.




(a) Service delay in ring

(b) Service delay in Internet2

Fig. 8. Service delay comparison of our proposed solution and alternative algorithms under different Zipf parameter $\alpha$: (a) shows the comparison results on the 10-node ring topology; (b) shows the comparison results on Internet2 topology. Our proposed algorithm outperforms the other two solutions. As $\alpha$ increases, both the joint and the allocation scheme yield less service delay, while the service delay of the single optimal routing method grows.

*5.4.2 Service Delay Under Different Popularity Distributions.* The parameter $\alpha$ of the Zipf distribution determines the shape of popularity distribution. Larger $\alpha$ indicates a larger fraction of requests on those very popular segments. We vary $\alpha$ from 0.7 to 1.5 to investigate the impact on service delay in Figure 8.

We obtain the following observations from this experiment. First, **as $\alpha$ increases (i.e., more requests on popular segments), both the joint and allocation schemes yield less service delay, while the service delay of the single optimal routing method grows.** This can be understood by examining its first derivative with respect to $\alpha$ as $\tau(\alpha)' = \sum_{n=1}^{N} C\lambda_n(I(x_n^\alpha - x_n)\log I - x_n(I^\alpha - I)\log x_n)$, where $C$ is a positive constant. By substituting $I = 20{,}000$, and $x_n = 220, \forall n = 1, 2, \ldots, N$ from the optimal routing baseline case into $f(\alpha)'$, we find that $f(\alpha)' > 0$ for $\alpha > 0$, $\alpha \neq 1$. On the other hand, in the optimal allocation baseline and our joint method, $x_n$ is set corresponding to $\lambda_n$. Therefore, for some nodes with large $\lambda_n$ and $x_n$, there is $(I(x_n^\alpha - x_n)\log I - x_n(I^\alpha - I)\log x_n) < 0$. This leads to their summation $f(\alpha)' < 0$. Second, we find that the cost saving for the single optimal routing scheme decreases as $\alpha$ increases. This implies that, when the popularity distribution has a small tail, the optimal resource allocation phase dominates the joint problem. However, the cost saving is least 1.7% and 3.9% in the ring network topology and the Internet2 network topology, respectively. Third, owing to the path diversity, the delay saving of the Internet2 topology is less than that of the 10-node ring topology.

## 6 CONCLUSION

This article investigated video distribution over ICN. We jointly considered the resource allocation for each ICN router to handle the cached video segments and the routing scheme for uncached contents. For each ICN router, we adopted the partial caching scheme to increase the cache hit ratio. Specifically, each ICN router needs to store only the highest bitrate version for a partial set of video segments and derive other lower bitrate versions based on local online transcoding. The benefit is gained on the loss of computing cost. For uncached contents, we designed a routing scheme to schedule the traffic load network-wide to reduce video delivery time over the ICN. Moreover, we built an optimization framework and an iterative approach to simultaneously find the optimums for both metrics. We analytically proved the convergence. Simulation results implicated that more

resources should be allocated to ICN routers with heavier request rates, and the routing scheme favors the shortest path to allocate more traffic. In comparison with the state-of-the-art algorithms (optimal routing and optimal allocation), our algorithm can reduce the delay by up to 29.4% and 5.4% in the Internet2 network topology respectively.

## A APPENDIX

### A.1 Proof of Lemma 1

We check the convexity of the objective function and each constraint.

First, we introduce an auxiliary variable $u_n = y_n^{1-\alpha}$. For the objective function (11), we can rewrite it as $\mathfrak{R}_{hit} \approx \sum_n^N \frac{u_n - 1}{I^{1-\alpha}-1}$. We check the second-order derivatives of $-\mathfrak{R}_{hit}$ as follows:

$$\frac{\partial^2(-\mathfrak{R}_{hit})}{\partial^2 x_n} = \frac{\partial^2(-\mathfrak{R}_{hit})}{\partial x_n u_n} = \frac{\partial^2(-\mathfrak{R}_{hit})}{\partial^2 u_n} = 0.$$

The Hessian matrix of $-\mathfrak{R}_{hit}$ is positive semi-definite.

Second, we rewrite constraint (12) as

$$\mathfrak{R}_{cache} = \sum_n^N (x_n(B_s - b_1) + y_n b_1) - \beta_1 = \sum_n^N \left(x_n(B_s - b_1) + u_n^{\frac{1}{1-\alpha}} b_1\right) - \beta_1.$$

The second-order derivatives are as follows:

$$\frac{\partial^2 \mathfrak{R}_{cache}}{\partial^2 x_n} = \frac{\partial^2 \mathfrak{R}_{cache}}{\partial x_n \partial u_n} = 0, \frac{\partial^2 \mathfrak{R}_{cache}}{\partial^2 u_n} = \frac{\alpha b_1}{(1-\alpha)^2} u_n^{\frac{2\alpha-1}{1-\alpha}} > 0,$$

The Hessian matrix of $\mathfrak{R}_{cache}$ is positive definite.

Third, we rewrite constraint (13) as

$$\mathfrak{R}_{trans} = \sum_{n=1}^N \lambda_n(\overline{b} - p_1 b_1)P_n^{Tr} - \beta_2 \approx \sum_{n=1}^N \lambda_n(\overline{b} - p_1 b_1)(1-p_1)\frac{u_n - x_n^{1-\alpha}}{I^{1-\alpha}-1} - \beta_2.$$

Therefore, we get the second-order derivatives as

$$\frac{\partial^2 \mathfrak{R}_{trans}}{\partial^2 x_n} = \lambda_n(\overline{b} - p_1 b_1)(1-p_1)\frac{\alpha(1-\alpha)}{I^{1-\alpha}-1} x_n^{-\alpha-1}, \frac{\partial^2 \mathfrak{R}_{trans}}{\partial x_n \partial u_n} = \frac{\partial^2 \mathfrak{R}_{trans}}{\partial^2 u_n} = 0,$$

Since $\frac{1-\alpha}{I^{1-\alpha}-1} > 0$, $\alpha > 0$, $\alpha \neq 1$, $\frac{\partial^2 \mathfrak{R}_{trans}}{\partial^2 x_n} > 0$. The Hessian matrix of $\mathfrak{R}_{trans}$ is positive definite.

Fourth, $O_e = \sum_{n=1}^N \sum_{k=1}^{K_n} \lambda_n \overline{b} P_n^{Miss} \pi_n^k \mathbb{I}_{e,R_{nk}} = \sum_{n=1}^N \sum_{k=1}^{K_n} \lambda_n \overline{b}(1 - P_n^{Hit}) \pi_n^k \mathbb{I}_{e,R_{nk}}$, which is also convex.

To summarize, all of the equations in problem **P1** are convex in $x_n$ and $u_n$. The optimization problem **P1** can be transformed to a convex optimization problem.

### A.2 Proof of Theorem 1

By checking the sufficient conditions for convergence of the Gauss-Seidel algorithm [3], we present the proof of Theorem 1. From [3], if $\overrightarrow{C}(x_n, u_n, \pi_n^k)$ and all the constraints are (1) bounded from below; (2) differentiable; (3) marginally convex in $x_n$, $u_n$, and $\pi_n^k$; and (4) jointly convex in $x_n$, $u_n$, and $\pi_n^k$, then it will converge to the minimum of $\overrightarrow{C}(x_n, u_n, \pi_n^k)$.

Based on the system formulation and Lemmas 1 and 2, the first three conditions are satisfied. Specifically, condition (1) is satisfied for $x_n \leq 0, y_n \leq 0, u_n \leq 0$, and $\pi_n^k \leq 0$ by definition. Condition (2) is satisfied because $\overrightarrow{C}(x_n, u_n, \pi_n^k)$ and all of the constraints are differentiable. Condition (3) is satisfied based on Lemmas 1 and 2.

Next, we show that the last condition is also satisfied. Constraints (12) and (13) are convex and independent on $\pi_n^k$. Constraint (6) is convex and independent on $x_n$ and $u_n$. Thereby, constraints (12), (13), and (6) are jointly convex in $x_n$, $u_n$, and $\pi_n^k$. Constraint (14) is a linear combination of $P_n^{Miss}$ and $\pi_n^k$. Since $P_n^{Miss}$ is convex in $x_n$ and $u_n$ and independent on $\pi_n^k$, and $\pi_n^k$ is convex and independent on $x_n$ and $u_n$, the constraint (14) is jointly convex. Similarly, the joint objective function $\overrightarrow{C}(x_n, u_n, \pi_n^k)$ is also jointly convex.

## REFERENCES


[1] Noor Abani, Torsten Braun, and Mario Gerla. 2017. Proactive caching with mobility prediction under uncertainty in information-centric networks. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*. ACM, 88–97.

[2] Mikhail Badov, Anand Seetharam, Jim Kurose, Victor Firoiu, and Soumendra Nanda. 2014. Congestion-aware caching and search in information-centric networks. In *Proceedings of the 1st International Conference on Information-centric Networking*. ACM, 37–46.

[3] Dimitri P. Bertsekas and John N. Tsitsiklis. 1989. *Parallel and Distributed Computation: Numerical Methods*. Vol. 23. Prentice Hall, Englewood Cliffs, NJ.

[4] Muhammad Bilal and Shin-Gak Kang. 2018. Network-coding approach for information-centric networking. *IEEE Systems Journal* 99 (2018), 1–10.

[5] Cisco. 2017. Cisco Visual Networking Index: Forecast and Methodology, 2016–2021. Retrieved December 21, 2018 from https://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf.

[6] Giuseppe Cofano, Luca De Cicco, Thomas Zinner, Anh Nguyen-Ngoc, Phuoc Tran-Gia, and Saverio Mascolo. 2017. Design and performance evaluation of network-assisted control strategies for HTTP adaptive streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications* 13, 3 (2017), 42.

[7] Ali Dabirmoghaddam, Maziar Mirzazad Barijough, and JJ Garcia-Luna-Aceves. 2014. Understanding optimal caching and opportunistic caching at the edge of information-centric networks. In *Proceedings of the 1st International Conference on Information-centric Networking*. ACM, 47–56.

[8] Ikram Ud Din, Suhaidi Hassan, Muhammad Khurram Khan, Mohsen Guizani, Osman Ghazali, and Adib Habbal. 2018. Caching in information-centric networking: Strategies, challenges, and future research directions. *IEEE Communications Surveys & Tutorials* 20, 2 (2018), 1443–1474.

[9] Suyong Eum, Kiyohide Nakauchi, Masayuki Murata, Yozo Shoji, and Nozomu Nishinaga. 2012. CATT: Potential based routing with content caching for ICN. In *Proceedings of the 2nd ICN Workshop on Information-centric Networking*. ACM, New York, NY, 49–54.

[10] Seyed Kaveh Fayazbakhsh, Yin Lin, and et al. 2013. Less pain, most of the gain: Incrementally deployable ICN. In *ACM SIGCOMM*. ACM, 147–158.

[11] Guanyu Gao, Han Hu, Yonggang Wen, and Cedric Westphal. 2017. Resource provisioning and profit maximization for transcoding in clouds: A two-timescale approach. *IEEE Transactions on Multimedia* 19, 4 (2017), 836–848.

[12] Guanyu Gao, Yonggang Wen, and Han Hu. 2017. QDLCoding: QoS-differentiated low-cost video encoding scheme for online video service. In *IEEE Conference on Computer Communications (INFOCOM'17)*. IEEE, 1–9.

[13] Reinhard Grandl, Kai Su, and Cedric Westphal. 2013. On the interaction of adaptive video streaming with content-centric networking. In *20th International Packet Video Workshop*. IEEE, 1–8.

[14] Shuo Guo, Haiyong Xie, and Guangyu Shi. 2012. Collaborative forwarding and caching in content centric networks. In *Proceedings of the International Conference on Research in Networking*. Springer, 41–55.

[15] Mohammad Hajimirsadeghi, Narayan B. Mandayam, and Alex Reznik. 2017. Joint caching and pricing strategies for popular content in information centric networks. *IEEE Journal on Selected Areas in Communications* 35, 3 (2017), 654–667.

[16] Han Hu, Yonggang Wen, Huanbo Luan, Tat-Seng Chua, and Xuelong Li. 2014. Toward multiscreen social TV with geolocation-aware social sense. *IEEE MultiMedia* 21, 3 (July 2014), 10–19.

[17] Han Hu, Yonggang Wen, and Dusit Niyato. 2017. Public cloud storage-assisted mobile social video sharing: A supermodular game approach. *IEEE Journal on Selected Areas in Communications* 35, 3 (2017), 545–556.

[18] Han Hu, Yongang Wen, and Dusit Niyato. 2017. Spectrum allocation and bitrate adjustment for mobile social video sharing: Potential game with online QoS learning approach. *IEEE Journal on Selected Areas in Communications* 35, 4 (2017), 935–948.

[19] Baixiang Huang, Anfeng Liu, Chengyuan Zhang, Naixue Xiong, Zhiwen Zeng, and Zhiping Cai. 2018. Caching joint shortcut routing to improve quality of service for information-centric networking. *Sensors* 18, 6 (2018), 1750.

[20] Internet2. 2016. Internet2 Network Advanced Layer 3 Service. Retrieved December 1, 2018 from https://www.internet2.edu/media/medialibrary/2016/03/11/I2-Network-Infrastructure-Topology-L3-201603.pdf.

[21] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard. 2009. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*. ACM, 1–12.

[22] Yichao Jin and Yonggang Wen. 2014. PAINT: Partial in-network transcoding for adaptive streaming in information centric network. In *Proceedings of IEEE/ACM International Symposium of Quality of Service (IWQoS'09)*. IEEE, 208–217.

[23] Yichao Jin, Yonggang Wen, Han Hu, and M.-J. Montpetit. 2014. Reducing operational costs in cloud social TV: An opportunity for cloud cloning. *IEEE Transactions on Multimedia* 16, 6 (Oct 2014), 1739–1751.

[24] Yichao Jin, Yonggang Wen, and Cedric Westphal. 2015. Optimal transcoding and caching for adaptive streaming in media cloud: An analytical approach. *IEEE Transactions on Circuits and Systems for Video Technology* 25, 12 (2015), 1914–1925.

[25] Yichao Jin, Yonggang Wen, and Cedric Westphal. 2015. Towards joint resource allocation and routing to optimize video distribution over future Internet. In *IEEE/IFIP Networking Conference*. IEEE, 150–158.

[26] Seng-Kyoun Jo, Lin Wang, Jussi Kangasharju, and Max Mühlhäuser. 2018. Green named data networking using renewable energy. In *Proceedings of the 9th International Conference on Future Energy Systems*. ACM, 414–416.

[27] Derek Kulinski and Jeff Burke. 2012. *NDN Video: Live and Prerecorded Streaming over NDN*. Technical Report. The NDN Project Team.

[28] Stefan Lederer, Christopher Mueller, Christian Timmerer, Cyril Concolato, Jean Le Feuvre, and Karel Fliegel. 2013. Distributed DASH dataset. In *Proceedings of the 4th ACM Multimedia Systems Conference*. ACM, 131–135.

[29] Stefan Lederer, Christopher Mueller, Christian Timmerer, and Hermann Hellwagner. 2014. Adaptive multimedia streaming in information-centric networks. *IEEE Network* 28, 6 (2014), 91–96.

[30] Yanhua Li, Haiyong Xie, Yonggang Wen, and Zhi-Li Zhang. 2013. Coordinating in-network caching in content-centric networks: Model and analysis. In *IEEE International Conference on Distributed Computing Systems (ICDCS'13)*. IEEE, 62–72.

[31] Muhammad Faran Majeed, Syed Hassan Ahmed, Siraj Muhammad, Houbing Song, and Danda B. Rawat. 2017. Multimedia streaming in information-centric networking: A survey and future perspectives. *Computer Networks* 125 (2017), 103–121.

[32] Deloitte. 2017. Media Consumer Survey 2017. Australian media and digital entertainment preferences. Retrieved December 21, 2018 from https://www2.deloitte.com/au/en/pages/technology-media-and-telecommunications/articles/media-consumer-survey-2017.html.

[33] Konstantin Miller, Abdel-Karim Al-Tamimi, and Adam Wolisz. 2017. QoE-based low-delay live streaming using throughput predictions. *ACM Transactions on Multimedia Computing, Communications, and Applications* 13, 1 (2017), 4.

[34] Ioannis Psaras, Wei Koong Chai, and George Pavlou. 2014. In-network cache management and resource allocation for information-centric networks. *IEEE Transactions on Parallel and Distributed Systems* 25, 11 (2014), 2920–2931.

[35] Jennifer Rexford. 2006. Route optimization in IP networks. In *Handbook of Optimization in Telecommunications*. Springer, 679–700.

[36] Dario Rossi and Giuseppe Rossini. 2012. On sizing CCN content stores by exploiting topological information. In *Proceeding of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 280–285.

[37] Giuseppe Rossini and Dario Rossi. 2014. Coupling caching and forwarding: Benefits, analysis, and implementation. In *Proceedings of the 1st International Conference on Information-centric Networking*. ACM, 127–136.

[38] Manolis Sifalakis, Basil Kohler, Christopher Christopher, and Christian Tschudin. 2014. An information centric network for computing the distribution of computations. In *Proceedings of the 1st International Conference on Information-centric Networking*. ACM, 137–146.

[39] Ashkan Sobhani, Abdulsalam Yassine, and Shervin Shirmohammadi. 2017. A video bitrate adaptation and prediction mechanism for HTTP adaptive streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications* 13, 2 (2017), 18.

[40] Kai Su and Cedric Westphal. 2014. On the benefit of information centric networks for traffic engineering. In *IEEE International Conference on Communications (ICC'17)*. IEEE, 3178–3184.

[41] Yi Sun, Seyed Kaveh Fayaz, Yang Guo, Vyas Sekar, Yun Jin, Mohamed Ali Kaafar, and Steve Uhlig. 2014. Trace-driven analysis of ICN caching algorithms on video-on-demand workloads. In *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*. ACM, 363–376.

[42] Christian Tschudin and Manolis Sifalakis. 2014. Named functions and cached computations. In *IEEE 11th Consumer Communications and Networking Conference (CCNC'14)*. IEEE, 851–857.

[43] Cedric Westphal, Christopher Mueller, Andrea Detti, Daniel Corujo, Jianping Wang, Marie-Jose Montpetit, Niall Murray, Shucheng LIU (Will), Stefan Lederer, Christian Timmerer, and Daniel Posch. 2016. Adaptive Video Streaming over Information-Centric Networking (ICN'16). RFC 7933, IRTF. http://www.rfc-editor.org/rfc/rfc7933.txt.

[44] Edmund Yeh, Tracey Ho, Ying Cui, Michael Burd, Ran Liu, and Derek Leong. 2014. VIP: A framework for joint dynamic forwarding and caching in named data networks. In *Proceedings of the 1st International Conference on Information-centric Networking*. ACM, 117–126.

[45] W.-P.K. Yiu, Xing Jin, and S.-H.G. Chan. 2007. VMesh: Distributed segment storage for peer-to-peer interactive video streaming. *IEEE Journal on Selected Areas in Communications* 25, 9 (2007), 1717–1731.

[46] Weiwen Zhang, Yonggang Wen, Jianfei Cai, and Dapeng Oliver Wu. 2014. Toward transcoding as a service in a multimedia cloud: Energy-efficient job-dispatching algorithm. *IEEE Transactions on Vehicular Technology* 63, 5 (2014), 2002–2012.

[47] Liang Zhou. 2016. Mobile device-to-device video distribution: Theory and application. *ACM Transactions on Multimedia Computing, Communications, and Applications* 12, 3 (2016), 38.

Received January 2018; revised October 2018; accepted October 2018