The Study of Dynamic Caching via State Transition Field - the Case of Time-Invariant Popularity

Jie Gao, *Member, IEEE*, Lian Zhao, *Senior Member, IEEE*, and Xuemin (Sherman) Shen, *Fellow, IEEE*

Abstract

This two-part paper investigates cache replacement schemes with the objective of developing a general model to unify the analysis of various replacement schemes and illustrate their features. To achieve this goal, we study the dynamic process of caching in the vector space and introduce the concept of state transition field (STF) to model and characterize replacement schemes. In the first part of this work, we consider the case of time-invariant content popularity based on the independent reference model (IRM). In such case, we demonstrate that the resulting STFs are static, and each replacement scheme leads to a unique STF. The STF determines the expected trace of the dynamic change in the cache state distribution, as a result of content requests and replacements, from any initial point. Moreover, given the replacement scheme, the STF is only determined by the content popularity. Using four example schemes including random replacement (RR) and least recently used (LRU), we show that the STF can be used to analyze replacement schemes such as finding their steady states, highlighting their differences, and revealing insights regarding the impact of knowledge of content popularity. Based on the above results, STF is shown to be useful for characterizing and illustrating replacement schemes. Extensive numeric results are presented to demonstrate analytical STFs and STFs from simulations for the considered example replacement schemes.

Index Terms

cache replacement policy, probabilistic caching, cache state transition, IRM, online caching, mobile edge caching.

J. Gao and X. Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, N2L 3G1, Canada (e-mail: {jie.gao, sshen}@uwaterloo.ca).

L. Zhao is with the Department of Electrical, Computer, and Biomedical Engineering, Ryerson University, Toronto, ON, M5B 2K3, Canada (e-mail: l5zhao@ryerson.ca).

I. Introduction

Caching has been attracting an increasing amount of attention in the research of wireless communications, especially in the context of mobile edge caching [1] and the joint study of communication, computation, and caching with the objective of deploying services close to mobile users [2], [3]. The research on the performance of caching in wireless communication systems may adopt various metrics. The focus can be decreasing the content delivery latency [4], alleviating congestion over the backhaul [5], reducing energy consumption [6], or a combination of the above [7]. While metrics can be different, the underlying caching performance is largely centered around one measurement, i.e., the cache hit ratio. Since a cache can only accommodate a limited portion of all contents, the cache hit ratio is determined by how the cached contents are selected and how they are updated.

Selecting the contents to be cached is relevant in the context of proactive caching. For example, an edge node can cache contents in advance during off-peak hours to reduce peak-hour network traffic load [8]- [10]. The key to proactive caching is adapting to unknown content popularity or network environment, usually leading to a Markov decision problem [11] or a learning problem [12].

Updating the cached contents is relevant in the context of online caching. Specifically, a cached content may be evicted and replaced by a new content whenever a cache miss occurs, which leads to a dynamic process that updates cached contents on the fly [13]. The guiding rule in updating the contents is referred to as a *cache replacement scheme*. Evidently, the cache replacement scheme has a significant impact on the performance of caching. In fact, even if the contents are cached proactively, cache replacement can still play an important roll in updating the cached contents while requests are being received.

Due to the importance of cache replacement schemes, related topics have been extensively studied in various scenarios [14]. Classic replacement schemes include first in first out (FIFO), least recently used (LRU), least frequently used (LFU), random replacement (RR), *etc.* and their variants. Some early works adopted simple probabilistic models with primitive assumptions on the request distribution [15] or focused on bounding the performance of the aforementioned schemes [16]. More recent works adopted Markov chains to model and analyze cache replacement schemes [17]- [20]. This class of studies generally focused on deriving the steady states of the aforementioned schemes and the mixing time of their underlying Markov chains [21] [22]. In

our previous work [23], we considered the problem in reverse and designed the Markov chain underlying the replacement scheme so that a target set of content caching probabilities can be achieved.

Most recent works in the communications field tended to evaluate existing replacement schemes in their considered network scenarios or propose new schemes that suit their specific objectives. Chang *et al.* studied the joint problem of cache replacement and bandwidth allocation in the scenario of peer-assisted video-on-demand systems and compared different cache replacements through simulations [24]. Fiore *et al.* developed a replacement scheme for boosting content diversity in a wireless ad hoc network based on the estimated content presence at peer nodes [25]. A least fresh first scheme was designed in [26] to maintain the freshness of cached data for the scenario of the Internet of Things based on named data networking. Two replacement schemes were proposed for the video-on-demand service in femtocells [27], the first of which exploits content access history for improving cache hit ratio while the second exploits information on user access delay to promote service fairness. Kamiyama *et al.* proposed a replacement scheme for content delivery networks based on the hop count from end users to the content server with an objective to reduce network traffic load [28]. Chattopadhyay *et al.* investigated content replacement based on the knowledge of cached contents at neighbor base stations for a cellular network with densely deployed base stations [29]. A similar scenario was studied in [30], in which the authors proposed replacement schemes that implicitly coordinate contents at caches over the network to maximize the overall hit ratio of the considered system.

While there has been abundant research on the topic of cache replacement, a model that can conveniently unify the analysis of different replacement schemes, characterize their features, and intuitively illustrate their differences is not yet available. The objective of this two-part paper is to develop such a model. Specifically, we have three targets. First, we aim to integrate cache replacement schemes under a unified general probabilistic cache replacement model and demonstrate this using several specific schemes as examples. Second, we target at studying the general cache replacement model from a novel perspective, the state transition field (STF), which characterizes replacement schemes in the vector space and captures the insights on their features. Third, we strive toward the goal of developing the model and methodology for studying cache replacements using the SFT.

The first part of this work focuses on the case when the content popularity is time-invariant while the second part investigates the scenario of time-varying content popularity [31]. Through

the two parts of this paper, we demonstrate that a replacement scheme corresponds to a unique state transition matrix, which in turn generates a unique STF, and the resulting STF jointly determines the performance of the replacement scheme with the content request statistics. Furthermore, although such an extension is not directly included, we provide the motivation, basic model, and methodology for studying the problem in reverse: given a performance target, can a replacement scheme be designed through determining the state transition matrix, which is in turn generated based on creating the STF according to the performance target and content request statistics?

The contributions of the first part are the followings.

First, we propose a general content replacement model based on probabilistic state transition as a unified model for cache replacement schemes. Unlike existing general model based on Markov chains, e.g., [21], we focus not just on the steady states but more on the dynamic change of the cache state distribution and describe this dynamic change in the vector space of state caching probabilities. Moreover, we introduce new ideas and results, such as the decomposition of state transition probability matrices based on contents and the mapping between state and content caching probabilities, to form a complete toolset for establishing our model.

Second, based on the aforementioned model, we introduce STF, which is a vector field defined over the state transition domain. We demonstrate that STF can characterize and illustrate cache replacement schemes. The STF determines the expected change of the dynamic cache state distribution just like an electromagnetic field determines the movement of a charged particle placed in it (although the STF can have more than 3 dimensions). Moreover, we show that the steady state of replacement schemes can be conveniently found based on the STF.

Third, we analyze the STF using four example replacement schemes of three types as case studies: RR, replace less popular (LP) and replace the least popular (TLP), and LRU. RR exploits no knowledge of content popularity, LP and TLP exploit perfect knowledge based on an assumption of perfect prediction, and LRU exploits imperfect knowledge from historical requests. We compare their STF and analyze the impact of the exploited knowledge on their steady states through the STF. Moreover, we conduct extensive simulations to generate the STF of the above example schemes to demonstrate the impact of replacement schemes and content popularity on the STF.

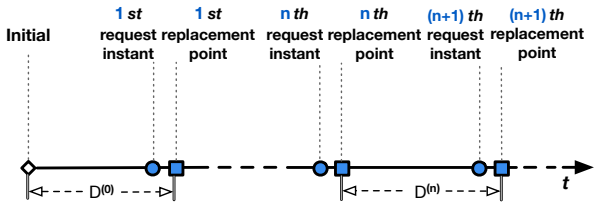


Fig. 1: Illustration of the timeline model. $\mathrm{D}^{(n)}$ represents the duration between the nth and the $(n+1)$th replacement points.

II. System Model

The scenario of N_c contents and a cache with size L is considered. The set of all contents is denoted by $\mathcal{C}$. Without loss of generality, we assume that all contents are of identical and unit size. We do not target at a specific scenario as the model can be applied to a cache located at a small-cell base station in a cellular network, a road side unit (RSU) in a vehicular network, or even a user device for D2D caching.

A. Content Request and Replacement

The fundamental assumption in the first part of this paper is that the requested contents at all instants, as integer random variables, are independent and identically distributed. This follows from the widely used independent reference model (IRM), a simplification of the actual request process that can be accurate with a large number of requesting users [21] or within a short time frame [32]. As the requested content follows a distribution that is time-invariant, the probability of content $l \in \mathcal{C}$ being requested can be denoted by υ_l. The probabilities $\{\upsilon_l\}_{\forall l}$ are organized into the request probability vector $\boldsymbol{\upsilon}$ and referred to as the content popularity.

If content l is requested but not being cached, it will be downloaded and, depending on the replacement scheme, may replace one cached content. It is assumed that the download and replacement can be completed before the next content request arrives at the cache.

The timeline of the considered dynamic caching is illustrated in Fig. 1. For simplicity of notation, we place a replacement point after each request regardless of whether a replacement actually happens or not. If a replacement occurs following the nth request, it is completed by the nth replacement point.

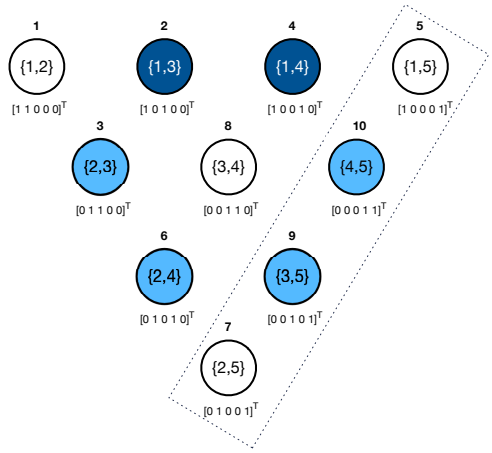


Fig. 2: An illustration of states with $N_{\mathrm{c}} = 5$ and $L = 2$. Each circle represents a state. The number above a circle represents the state ID, and the set inside a circle represents the set of cached contents in that state. For example, state 7 caches contents 2 and 5.

B. Cache State

The *cache state* is introduced to describe the combination of cached contents. There are $N_{\mathrm{s}} = \binom{N_{\mathrm{c}}}{L}$ different possible combinations of cached contents, corresponding to N_{s} caching states. The set of all cache states is denoted by $\mathcal{S}$. The set of contents cached in state k is denoted by $\mathcal{C}_k$. The cache state vector for state k is defined as a $N_{\mathrm{c}} \times 1$ vector with elements determined as follows:

$$\mathbf{s}_k(l) = \begin{cases} 1, & \text{if } \ l \in \mathcal{C}_k, \\ 0, & \text{if } \ l \notin \mathcal{C}_k, \end{cases} \quad \forall l \in \mathcal{C}, \forall k \in \mathcal{S}, \tag{1}$$

where the lth element of vector $\mathbf{s}_k$ corresponds to the lth content. An example with $N_{\mathrm{c}} = 5$ and $L = 2$ is illustrated in Fig. 2. In this example, there are $\binom{5}{2} = 10$ states. Each circle in the figure represents a state, while the number above the circle represents the state ID. The set given in the circle of state k is the set of cached contents in state k, i.e., $\mathcal{C}_k$, and the vector beneath state k is $\mathbf{s}_k$. For example, state 7 caches contents $\{2, 5\}$ and is represented by the cache state vector $\mathbf{s}_7 = [0\,1\,0\,0\,1]^{\mathrm{T}}$, where $\cdot^{\mathrm{T}}$ stands for transpose, given beneath the circle of state 7 in Fig. 2.

A state is a neighbor of state k if its cached contents differ from those cached in state k by just one element. The set of neighbors of state k is denoted as $\mathcal{H}_k$. For any content $l \notin \mathcal{C}_k$, a content-l neighbor of state k is a neighboring state that caches l. The set of content-l neighboring

states of state k is denoted as $\mathcal{H}_{k,l}$. Using Fig. 2 and state 8 as an example, $\mathcal{H}_8$ is the set of all colored states, and $\mathcal{H}_{8,1}$ is the two states with deep color.

C. State and Content Caching Probabilities

The cached contents and the cache state remain constant in the durations between consecutive replacement points (shown in Fig. 1). The state caching probability (SCP) for state k and the nth duration, denoted by $\eta_k^{(n)}$, is the probability that the cache is in state k in the nth duration. The content caching probability (CCP) for content l and the nth duration, denoted by $\lambda_l^{(n)}$, is the probability that the content l is cached in the nth duration.

Define the SCP vector $\boldsymbol{\eta}^{(n)}$ and CCP vector $\boldsymbol{\lambda}^{(n)}$ such that $\boldsymbol{\eta}^{(n)}(k) = \eta_k^{(n)}$ and $\boldsymbol{\lambda}^{(n)}(l) = \lambda_l^{(n)}$. Evidently, $\mathbf{1}^{\mathrm{T}}\boldsymbol{\eta}^{(n)} = 1$ and $\mathbf{1}^{\mathrm{T}}\boldsymbol{\lambda}^{(n)} = L$. Based on the time line in Fig. 1, the SCP and the CCP vectors at the instant of the $(n+1)$th request are $\boldsymbol{\eta}^{(n)}$ and $\boldsymbol{\lambda}^{(n)}$, respectively.

The SCP and the CCP are connected through cache states. Using Fig. 2 as an example, the probability that content 5 is cached is equal to the sum of the probabilities that the states in the dotted box are cached. Define a cache state matrix $\mathbf{C}_{\mathrm{s}} = [\mathbf{s}_1, \ldots, \mathbf{s}_{N_{\mathrm{s}}}]$. In general, the relation between the SCP $\boldsymbol{\eta}^{(n)}$ and CCP $\boldsymbol{\lambda}^{(n)}$ is given by:

$$\boldsymbol{\lambda}^{(n)} = \mathbf{C}_{\mathrm{s}}\boldsymbol{\eta}^{(n)}. \tag{2}$$

D. Cache Hit Probability

Given that content l is being requested at the $(n+1)$th request, the conditional instantaneous cache hit probability is $\lambda_l^{(n)}$. The instantaneous cache hit probability at the $(n+1)$th request, denoted by $\gamma^{(n+1)}$ is given by:

$$\gamma^{(n+1)} = \boldsymbol{\upsilon}^{\mathrm{T}}\boldsymbol{\lambda}^{(n)}. \tag{3}$$

The symbols used in this paper are listed in Table I. Throughout the paper, we use lower-case bold letters for vectors, upper-case bold letters for matrices, and calligraphic letters for sets. The superscript $(\cdot)^{(n)}$ is used on letters related to the nth request or replacement. Greek letters are used to represent various probabilities. The indexes m and k are used to denote cache states, while the indexes l and q are used to denote contents.

TABLE I: List of Symbols

N_{c}	The number of all contents
N_{s}	The number of all cache states
L	The cache size limit
$\mathcal{C}$	The set of all contents, i.e., $\{1,\ldots,N_{\mathrm{c}}\}$
$\mathcal{S}$	The set of all cache states, i.e., $\{1,\ldots,N_{\mathrm{s}}\}$
$\mathcal{S}_l$	The set of all cache states that cache content l
$\mathbf{s}_k$	The kth cache state vector
$\mathcal{C}_k$	The set of contents cached in state k
$\mathbf{C}_{\mathrm{s}}$	The cache state matrix, i.e., $[\mathbf{s}_1,\ldots,\mathbf{s}_{N_{\mathrm{s}}}]$
$\mathcal{H}_k$	The set of all neighbors of state k
$\mathcal{H}_{k,l}$	The set of all content-l neighbors of state k
$e(k,m)$	The unique element in the set $\mathcal{C}_k-\mathcal{C}_m$, where $m\in\mathcal{H}_k$
υ_l	The request probability of content l
$\boldsymbol{\upsilon}$	The content request probability vector, i.e., $[\upsilon_1,\ldots,\upsilon_{N_{\mathrm{c}}}]^{\mathrm{T}}$
$\phi_{l,q,k}$	The conditional probability that content l replaces content q given that cache is in state k and content l is requested
$\boldsymbol{\Theta}$	The state transition probability matrix
$\boldsymbol{\Theta}_l$	The conditional state transition probability matrix given that content l is requested
$\boldsymbol{\Theta}(m,k)$	The probability of transitioning from state k to state m
$\boldsymbol{\Theta}_l(m,k)$	The probability of transitioning from state k to state m given that content l is requested
$\eta_k^{(n)}$	The SCP for state k in the duration from the nth to the $(n+1)$th replacement
$\boldsymbol{\eta}^{(n)}$	The SCP vector in the duration from the nth to the $(n+1)$th replacement, i.e, $[\eta_1^{(n)},\ldots,\eta_{N_{\mathrm{s}}}^{(n)}]^{\mathrm{T}}$
$\lambda_l^{(n)}$	The CCP for content l in the duration from the nth to the $(n+1)$th replacement
$\boldsymbol{\lambda}^{(n)}$	The CCP vector in the duration from the nth to the $(n+1)$th replacement, i.e., $[\lambda_1^{(n)},\ldots,\lambda_{N_{\mathrm{c}}}^{(n)}]^{\mathrm{T}}$
$\gamma^{(n)}$	The instantaneous cache hit probability at the nth request
$\mathbf{u}(\boldsymbol{\eta})$	The state transition field at $\boldsymbol{\eta}$
$\mathbf{u}_l(\boldsymbol{\eta})$	The content-l state transition field at $\boldsymbol{\eta}$
$u_{m,l}(\boldsymbol{\eta})$	The mth element of the state transition field at $\boldsymbol{\eta}$
$u_{m,l}(\boldsymbol{\eta})$	The mth element of the content-l state transition field at $\boldsymbol{\eta}$

III. General Content Replacement Model and State Transition Field

If the cache is at state k while content $l\notin\mathcal{C}_k$ is requested, the cache downloads content l and decides whether to replace a cached content with content l. In the general model, the probability

of replacing content q with content l when the cache is at state k is denoted by $\phi_{l,q,k}$, for any $q \in \mathcal{C}_k$ and $l \notin \mathcal{C}_k$. For each state, there are $L(N_\mathrm{c} - L)$ possible replacements.

A. General Cache State Transition Model

A content replacement triggers a cache state transition. For neighboring states k and m which satisfies $m \in \mathcal{H}_{k,l}$ and $k \in \mathcal{H}_{m,q}$, replacing content q with l triggers a transition from state k to state m. The conditional cache state transition probabilities given that content l is requested can be organized into the following matrix $\mathbf{\Theta}_l$:

$$\mathbf{\Theta}_l(m,k) = \begin{cases} 1, & \text{if } k = m \text{ and } l \in \mathcal{C}_k, \\ 1 - \sum\limits_{m' \in \mathcal{H}_{k,l}} \phi_{l,e(k,m'),k}, & \text{if } k = m \text{ and } l \notin \mathcal{C}_k, \\ \phi_{l,e(k,m),k}, & \text{if } m \in \mathcal{H}_{k,l}, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where $e(k,m)$ denotes the unique content that is cached by state k but not state m given that $k \in \mathcal{H}_m$. Accordingly, the overall cache state transition probability matrix in the general case is given by:

$$\mathbf{\Theta} = \sum_{l \in \mathcal{C}} \upsilon_l \mathbf{\Theta}_l. \tag{5}$$

From the definition of the SCP vector $\boldsymbol{\eta}^{(n)}$ and state transition probability matrix $\mathbf{\Theta}$, it can be seen that:

$$\boldsymbol{\eta}^{(n)} = \mathbf{\Theta} \boldsymbol{\eta}^{(n-1)}. \tag{6}$$

It is worth mentioning that the model can be extended to the scenario in which each content request (and replacement) involves multiple contents. In such case, assuming that each request is for a block of B contents ($B < L$), there are $N_\mathrm{B} = \binom{N_\mathrm{c}}{B}$ different blocks. Then, eq. (5) can be extended as follows:

$$\mathbf{\Theta}^\mathrm{B} = \sum_{b=1}^{N_\mathrm{B}} \upsilon_b^\mathrm{B} \mathbf{\Theta}_b^\mathrm{B}, \tag{7}$$

where υ_b^B is the probability that the bth block is requested, and $\mathbf{\Theta}_b^\mathrm{B}$ is the conditional cache state transition probabilities given that block b is requested. The size of $\mathbf{\Theta}_b^\mathrm{B}$ remains $N_\mathrm{s} \times N_\mathrm{s}$. However, for any given state, e.g., state k, the set of its neighbors $\mathcal{H}_k$ will contain more states under block replacement, and the set of its content-l neighbors $\mathcal{H}_{k,l}$ will be replaced by a set of block-b neighbors. The extension is straightforward and the details are omitted here.

B. STF

Denote the general SCP without specifying any time instant as $\boldsymbol{\eta}$. Consider $\boldsymbol{\eta}$ as a point in the N_{s}-dimensional vector space. Driving by the requests and replacements, $\boldsymbol{\eta}$ varies in the following domain:

$$\mathcal{D}=\left\{\left(\eta_1,\ldots,\eta_{N_{\mathrm{s}}}\right)\middle|0\leq\eta_k\leq 1,\forall k\in\mathcal{S};\sum_k\eta_k=1\right\}. \tag{8}$$

The expected 'movement' of $\boldsymbol{\eta}$ in $\mathcal{D}$ after the nth replacement point, assuming a replacement actually happens, is characterized by $\boldsymbol{\eta}^{(n)}-\boldsymbol{\eta}^{(n-1)}$. This difference, in turn, is determined by three factors:

- the current position of $\boldsymbol{\eta}$ in $\mathcal{D}$, i.e., the value of $\boldsymbol{\eta}^{(n-1)}$
- the content popularity $\boldsymbol{\upsilon}$
- the state transition probability matrix $\boldsymbol{\Theta}$,

while $\boldsymbol{\Theta}$ is determined by the replacement scheme and generally dependent on $\boldsymbol{\upsilon}$ (and such dependence is shown in eq. (5)).

Define the STF at the point $\boldsymbol{\eta}^{(n-1)}$ using the aforementioned difference:

$$\mathbf{u}(\boldsymbol{\eta}^{(n-1)})=\boldsymbol{\eta}^{(n)}-\boldsymbol{\eta}^{(n-1)}. \tag{9}$$

Substituting eq. (6) into eq. (9), it follows that:

$$\mathbf{u}(\boldsymbol{\eta}^{(n-1)})=\boldsymbol{\Theta}\boldsymbol{\eta}^{(n-1)}-\boldsymbol{\eta}^{(n-1)}. \tag{10}$$

The STF is a vector field defined over the domain $\mathcal{D}$. It can be seen that understanding the STF can provide insight into the design and performance analysis of replacement schemes. Similar to a magnetic or electric field, the STF can vary in direction and strength at different points in the domain (although the STF exists mathematically but not physically).

In the definition eq. (9), the $\boldsymbol{\eta}^{(n-1)}$ in the brackets specifies a point in the domain $\mathcal{D}$. If the STF is known at all points in $\mathcal{D}$, then a path can be identified from any initial point, as illustrated in Fig. 3, the end of which gives the steady state of the replacement scheme while the number of steps in the path reflects the time for the underlying Markov chain to attain its stationary state from that initial point. Different replacement schemes yield different STFs, and the impact is conveyed through $\boldsymbol{\Theta}$. Therefore, the STF is a complete characterization of replacement schemes.

It is worth noting that the STF does not change over time under the IRM in general, as $\boldsymbol{\upsilon}$ and $\boldsymbol{\Theta}$ are both constant.

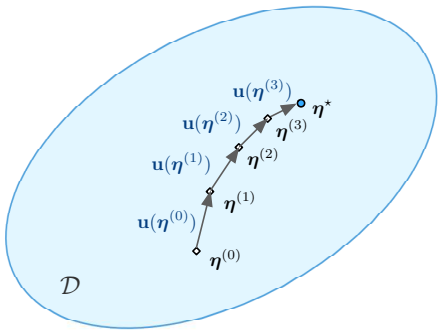


Fig. 3: An illustration of STF at four points, i.e., $\boldsymbol{\eta}^{(0)}$ to $\boldsymbol{\eta}^{(3)}$. The end point $\boldsymbol{\eta}^{\star}$ represents the steady state, at which the STF diminishes to an all-zero vector.

C. Content-specific STF

The STF can be decomposed. Define:

$$\mathbf{u}_l(\boldsymbol{\eta}^{(n-1)}) = \boldsymbol{\Theta}_l\boldsymbol{\eta}^{(n-1)} - \boldsymbol{\eta}^{(n-1)}. \tag{11}$$

It follows that:

$$\sum_{l\in\mathcal{C}} \upsilon_l \mathbf{u}_l(\boldsymbol{\eta}^{(n-1)}) = \sum_{l\in\mathcal{C}} \upsilon_l \boldsymbol{\Theta}_l\boldsymbol{\eta}^{(n-1)} - \boldsymbol{\eta}^{(n-1)} = \mathbf{u}(\boldsymbol{\eta}^{(n-1)}), \tag{12}$$

where the last step uses eq. (5). Accordingly, $\mathbf{u}_l(\boldsymbol{\eta}^{(n-1)})$ can be considered as the content-specific STF that represents the 'movement' of $\boldsymbol{\eta}$ from the point $\boldsymbol{\eta}^{(n-1)}$ after content l is requested. The superposition of all content-specific STFs, weighted by the corresponding content popularity, yields the overall STF.

It is not difficult to see that the following equalities hold:

$$\mathbf{1}^{\mathrm{T}}\mathbf{u}_l(\boldsymbol{\eta}^{(n-1)}) = 0,\ \forall l \in \mathcal{C},\ \forall \boldsymbol{\eta}^{(n-1)} \in \mathcal{D} \tag{13}$$

$$\mathbf{1}^{\mathrm{T}}\mathbf{u}(\boldsymbol{\eta}^{(n-1)}) = 0,\ \forall \boldsymbol{\eta}^{(n-1)} \in \mathcal{D}. \tag{14}$$

IV. State Transition Matrices of Specific Replacement Schemes

In this section, we demonstrate how four specific replacement schemes, i.e., RR, LP, TLP, and LRU, fit into the general content replacement model in the preceding section. As the impact of replacement schemes on the STFs is conveyed through the state transition matrix $\boldsymbol{\Theta}$, the focus will be on finding $\boldsymbol{\Theta}$ for the considered schemes.

The four replacement schemes can be categorized into three groups based on the content popularity information that they exploit.

- RR does not use any content popularity information;
- Both LP and TLP rely on the prediction of content popularity, and a perfect prediction will be assumed.
- LRU exploits imperfect content popularity information from request history, i.e., the information of recent content requests.

The impact of the difference in the exploited content popularity information on the STF will be presented in subsequent sections of this paper.

A. RR

For RR, the conditional content replacement probability $\phi_{l,q,k}$ reduces to a constant:

$$\phi_{l,q,k} = \phi \in (0, 1/L], \quad \forall q \in \mathcal{C}_k, l \notin \mathcal{C}_k. \tag{15}$$

Accordingly, the conditional state transition probability matrix $\mathbf{\Theta}_l$ is given by:

$$\mathbf{\Theta}_{\mathrm{RR},l}(m,k) = \begin{cases} 1, & \text{if } \ l \in \mathcal{C}_k \ \text{ and } \ k = m, \\ 1 - L\phi, & \text{if } \ l \notin \mathcal{C}_k \ \text{ and } \ k = m, \\ \phi, & \text{if } \ m \in \mathcal{H}_{k,l}, \\ 0, & \text{otherwise}, \end{cases} \tag{16}$$

i.e., the probabilities of content l replacing a cached content and no replacement are $L\phi$ and $1 - L\phi$, respectively.

The overall state transition probability matrix $\mathbf{\Theta}_{\mathrm{RR}}$ is given by:

$$\mathbf{\Theta}_{\mathrm{RR}}(m,k) = \begin{cases} 1 - L\phi \sum\limits_{l \notin \mathcal{C}_k} \upsilon_l, & \text{if } \ k = m, \\ \phi \upsilon_{e(m,k)}, & \text{if } \ m \in \mathcal{H}_k, \\ 0, & \text{otherwise}. \end{cases} \tag{17}$$

B. LP

Denote the predicted content popularity by $\tilde{\boldsymbol{\upsilon}}$. Using LP, the requested content $l \notin \mathcal{C}_k$ may replace a cached content q in state k if $\tilde{\upsilon}_l > \tilde{\upsilon}_q$, i.e., the requested content is more popular. The

conditional state transition probability is given by:

$$\begin{aligned}&\boldsymbol{\Theta}_{\mathrm{LP},l}(m,k)\\&=\begin{cases}1, & \text{if } l\in\mathcal{C}_k \text{ and } k=m,\\ 1-\alpha, & \text{if } l\notin\mathcal{C}_k,\ k=m, \text{ and } \tilde{v}_l>\tilde{v}_q,\\ \alpha\phi_{l,q,k}, & \text{if } m\in\mathcal{H}_{k,l}, k\in\mathcal{H}_{m,q}, \text{ and } \tilde{v}_l>\tilde{v}_q,\\ 0, & \text{otherwise,}\end{cases}\end{aligned} \tag{18}$$

where α is a parameter that controls the probability of a replacement.

The conditional replacement probability, assuming that $\tilde{v}_l > \tilde{v}_q$, is set to be proportional to $\tilde{v}_l - \tilde{v}_q$, as follows:

$$\phi_{l,q,k} = \frac{\tilde{v}_l - \tilde{v}_q}{\sum\limits_{t\in\mathcal{C}_{k,l}^{\downarrow}} (\tilde{v}_l - \tilde{v}_t)}, \tag{19}$$

where

$$\mathcal{C}_{k,l}^{\downarrow} = \{t | t\in\mathcal{C}_k, \tilde{v}_t < \tilde{v}_l\}. \tag{20}$$

Order the states based on $\sum_{t\in\mathcal{C}_k}\tilde{v}_t$, i.e., the summation of the predicted content request probability of each state, in a non-decreasing order. Then, it can be shown that the state transition matrix $\boldsymbol{\Theta}_{\mathrm{LP}}$ becomes a lower-triangular matrix:

$$\begin{aligned}&\boldsymbol{\Theta}_{\mathrm{LP}}(m,k)\\&=\begin{cases}\sum\limits_{q\in\mathcal{C}_k} v_q+\sum\limits_{l\in\bar{\mathcal{C}}_k^{\downarrow}} v_l+\sum\limits_{l\in\bar{\mathcal{C}}_k^{\uparrow}} v_l(1-\alpha), & \text{if } m=k,\\ \alpha v_{e(m,k)}\phi_{e(m,k),e(k,m),k}, & \text{if } m>k \text{ and } m\in\mathcal{H}_k,\\ 0, & \text{otherwise.}\end{cases}\end{aligned} \tag{21}$$

in which

$$\bar{\mathcal{C}}_k^{\downarrow} = \left\{ l \mid l\notin\mathcal{C}_k, \tilde{v}_l \le \min_{t\in\mathcal{C}_k}\{\tilde{v}_t\} \right\}, \tag{22a}$$

$$\bar{\mathcal{C}}_k^{\uparrow} = \left\{ l \mid l\notin\mathcal{C}_k, \tilde{v}_l > \min_{t\in\mathcal{C}_k}\{\tilde{v}_t\} \right\}. \tag{22b}$$

C. TLP

Denote the least popular content of state k based on the prediction by $q^{\dagger}(k)$, i.e.,

$$q^{\dagger}(k) = \operatorname*{argmin}_{t\in\mathcal{C}_k}\{\tilde{v}_t\}. \tag{23}$$

Using TLP, the requested content $l \notin \mathcal{C}_k$ can only replace $q^\dagger(k)$ when the cache is in state k, and the replacement can happen only if $\tilde{v}_l > \tilde{v}_{q^\dagger(k)}$. The conditional state transition probability is given by:

$$\begin{aligned}&\boldsymbol{\Theta}_{\mathrm{TLP},l}(m,k)\\&=\begin{cases}1, & \text{if } l\in\mathcal{C}_k \text{ and } k=m,\\ 1-\phi_{l,q^\dagger(k),k}, & \text{if } l\notin\mathcal{C}_k \text{ and } k=m, \text{ and } \tilde{v}_l>\tilde{v}_{q^\dagger(k)},\\ \phi_{l,q^\dagger(k),k}, & \text{if } m\in\mathcal{H}_{k,l}, k\in\mathcal{H}_{m,q^\dagger(k)}, \text{and } \tilde{v}_l>\tilde{v}_{q^\dagger(k)},\\ 0, & \text{otherwise.}\end{cases}\end{aligned} \tag{24}$$

Two choices of the replacement probability $\phi_{l,q^\dagger(k),k}$ are considered when $\tilde{v}_l > \tilde{v}_{q^\dagger(k)}$: $\phi_{l,q^\dagger(k),k} = 1$ and $\phi_{l,q^\dagger(k),k} = \tilde{v}_l - \tilde{v}_{q^\dagger(k)}$. In the first case, the replacement always occurs, and the TLP in such case will be referred to as TLP-A. In the second case, the replace occurs probabilistically, and the the TLP in such case will be referred to as TLP-P. Intuitively, TLP-A would lead to faster convergence while TLP-P could be useful when each replacement incurs a replacement cost.

Order the states based on $\sum_{t\in\mathcal{C}_k}\tilde{v}_t$, i.e., the summation of the predicted content request probability of each state, in a non-decreasing order. Then, the state transition matrix $\boldsymbol{\Theta}_{\mathrm{TLP}}$ also becomes a lower-triangular matrix:

$$\begin{aligned}&\boldsymbol{\Theta}_{\mathrm{TLP}}(m,k)\\&=\begin{cases}\sum\limits_{q\in\mathcal{C}_k} v_q+\sum\limits_{l\in\bar{\mathcal{C}}_k^{\downarrow}} v_l+\sum\limits_{l\in\bar{\mathcal{C}}_k^{\uparrow}} v_l(1-\phi_{l,q^\dagger(k),k}), & \text{if } m=k,\\ v_{e(m,k)}\phi_{e(m,k),q^\dagger(k),k}, & \text{if } m>k \text{ and } e(k,m)=q^\dagger(k),\\ 0, & \text{otherwise.}\end{cases}\end{aligned} \tag{25}$$

D. LRU

When the cache is in state k while content l is requested, the conditional state transition probability matrix $\boldsymbol{\Theta}_l$ is given by:

$$\boldsymbol{\Theta}_{\mathrm{LRU},l}(m,k)=\begin{cases}1, & \text{if } \ l\in\mathcal{C}_k \ \text{ and } \ k=m,\\ \rho^{\mathrm{LRU}}_{e(k,m)|k}, & \text{if } \ m\in\mathcal{H}_{k,l},\\ 0, & \text{otherwise.}\end{cases} \tag{26}$$

where $\rho^{\mathrm{LRU}}_{e(k,m)|k}$ represents the conditional probability that content $e(k,m)$ is the least recently used content given that the cache is in state k. The probability $\rho^{\mathrm{LRU}}_{e(k,m)|k}$ can be found, as a

simplified special case under IRM, based on Lemma 1 in the second part of this two-part paper, which addresses the more general case of time-varying content popularity [31].

The overall state transition probability matrix $\mathbf{\Theta}_{\mathrm{LRU}}$ is given by:

$$\mathbf{\Theta}_{\mathrm{LRU}}(m,k) = \begin{cases} \sum\limits_{l \in \mathcal{C}_k} \upsilon_l, & \text{if } \ k = m, \\ \upsilon_{e(m,k)} \rho^{\mathrm{LRU}}_{e(k,m)|k}, & \text{if } \ m \in \mathcal{H}_k, \\ 0, & \text{otherwise.} \end{cases} \tag{27}$$

Note that, unlike RR and LRU, LP and TLP are not practical replacement schemes. However, the latter two are considered here for the purpose of analyzing what the STF of a replacement scheme would become in the ideal case with perfect content popularity information, as a comparison to the cases with no and imperfect content popularity information (e.g., RR and LRU, respectively).

V. STF based Analysis for Cache Replacement Under Time-invariant Content Popularity

In this section, we analyze specific replacement schemes using the STF to demonstrate that analysis based on STF can characterize the features of different replacement schemes and reveal insights regarding their steady states.

A. RR

Using the definition of content-specific STF in eq. (11) and the state transition probability matrix of RR in eq. (16), it can be shown that the mth element of the content-specific STF at $\boldsymbol{\eta}$ is given by:

$$u_{m,l,\mathrm{RR}}(\boldsymbol{\eta}) = \begin{cases} \phi \sum\limits_{\{k | m \in \mathcal{H}_{k,l}\}} \eta_k, & \text{if } l \in \mathcal{C}_m, \\ -L\phi\eta_m, & \text{otherwise.} \end{cases} \tag{28}$$

Using the STF in eq. (28), the following result becomes straightforward.

Theorem 1: The steady state of RR, denoted by $\boldsymbol{\eta}^\star$, is independent on the parameter ϕ and satisfies the following property:

$$\eta^\star_m \sum_{l \notin \mathcal{C}_m} \upsilon_l = \frac{1}{L} \sum_{k \in \mathcal{H}_m} \eta^\star_k \upsilon_{e(m,k)}, \forall m \in \mathcal{S}. \tag{29}$$

Proof: See Section A in Appendix.

The property in Theorem 1 can be used to obtain a closed-form expression of the steady state. Define N_{s} vectors, one for each state, so that

$$\mathbf{a}_m(k) = \begin{cases} \sum\limits_{l \notin \mathcal{C}_m} \upsilon_l, & \text{if } k = m, \\ -\frac{1}{L}\upsilon_{e(m,k)}, & \text{if } m \in \mathcal{H}_k, \\ 0, & \text{otherwise.} \end{cases} \tag{30}$$

where $\mathbf{a}_m(k)$ represents the kth element of the vector for the mth state. Then, $N_{\mathrm{s}} - 1$ out of the N_{s} vectors are linearly independent. Define matrix $\mathbf{A}$ as follows:

$$\mathbf{A} = \left[\mathbf{a}_1, \ldots, \mathbf{a}_{N_{\mathrm{s}}-1}, \mathbf{1}\right]^{\mathrm{T}}, \tag{31}$$

where $\mathbf{1}$ is an all-one vector. Then, the steady state $\boldsymbol{\eta}^\star$ can be given by

$$\boldsymbol{\eta}^\star = \mathbf{A}^{-1}\mathbf{g}, \tag{32}$$

in which $\mathbf{g} = [0, \ldots, 0, 1]^{\mathrm{T}}$ is the vector that has 0 as its first $N_{\mathrm{s}} - 1$ elements and 1 as its last element.

Evidently, the steady state of RR does not maximize cache hit probability as RR does not exploit any content popularity information. The property in Theorem 1 characterizes the steady state of RR. Specifically, eq.(29) shows that the steady state of RR achieves such balance that, if a randomly selected cached content is to be replaced by a random content not cached, the resulting expected cache miss probability due to this replacement should be equal to the cache miss ratio of the steady state without any replacement.

The rate of convergence of a finite-state ergodic Markov chain is decided by the second largest eigenvalue of its transition probability matrix [33]. Specifically, it holds that [34]:

$$\|\boldsymbol{\Theta}^t\boldsymbol{\eta}^{(0)} - \boldsymbol{\eta}^\star(\boldsymbol{\Theta})\|_2 \leq d_2^t(\boldsymbol{\Theta})\|\boldsymbol{\eta}^{(0)}\|_2 \tag{33}$$

for any initial state distribution $\boldsymbol{\eta}^{(0)}$, where $\boldsymbol{\Theta}$ represents any ergodic Markov chain, $\boldsymbol{\eta}^\star(\boldsymbol{\Theta})$ represents the corresponding steady state, $d_2(\boldsymbol{\Theta})$ represents the second largest eigenvalue of $\boldsymbol{\Theta}$, and t is the number of steps since the initial point. While it is generally impossible to derive an eigenvalue of an arbitrary transition matrix $\boldsymbol{\Theta}$ in closed-form, the bounds on the second largest eigenvalue of a reversible transition matrix can be estimated [35]. The STF provides another intuitive perspective for analyzing the rate of convergence. In the case of RR, a larger ϕ implies stronger STF while the direction of the STF at all points remains the same. Therefore, a larger ϕ generally leads to a shorter mixing time.

B. LP and TLP

Note that, in practice, the L most popular contents can be placed in the cache from the beginning without using LP or TLP for replacements if the content popularity is known. However, as we intend to analyze the impact of content popularity information adopted by a replacement scheme on the path of state cache distribution starting from an arbitrary state, the analysis of LP and TLP is of interest.

For LP and TLP, the steady state is straightforward. Sort the contents based on a nondecreasing order of their predicted popularity so that $\tilde{v}_l \geq \tilde{v}_q$ if $l \geq q$. Sort the states based on $\sum_{t\in\mathcal{C}_k} \tilde{v}_t$, i.e., the summation of the predicted content request probability of each state, in a non-decreasing order. Then, the L least popular contents are cached in state 1, and the L most popular contents are cached in state N_s.

Lemma 1: The steady state for both LP and TLP is $\boldsymbol{\eta}^\star = [0, \ldots, 0, 1]$.

The proof is straightforward given that, for any $k \in \mathcal{S}$, the following two facts hold: 1). State k can only transition to state m if $m > k$; and 2). State k transitions to at least one neighboring state in $\mathcal{H}_k$ with a positive probability. The two observations can be made based on eq. (21) and eq. (25).

Compared to the steady state of RR, the result in Lemma 1 reflects the impact of exploiting content popularity information on the steady state of a replacement scheme.

Since both $\boldsymbol{\Theta}_{\mathrm{LP}}$ and $\boldsymbol{\Theta}_{\mathrm{TLP}}$ are lower-triangular matrices, the eigenvalues of $\boldsymbol{\Theta}_{\mathrm{LP}}$ and $\boldsymbol{\Theta}_{\mathrm{TLP}}$ are their respective diagonal elements. Evidently, neither of $\boldsymbol{\Theta}_{\mathrm{LP}}$ and $\boldsymbol{\Theta}_{\mathrm{TLP}}$ is ergodic. Nevertheless, the second largest eigenvalue of both falls in $(0, 1)$ in both cases, and the result in eq. (33) still holds for $\boldsymbol{\Theta}_{\mathrm{LP}}$ and $\boldsymbol{\Theta}_{\mathrm{TLP}}$. The largest eigenvalue is 1 in both cases. The second largest eigenvalue, which determines the mixing time of LP and TLP, is given by the following result.

Lemma 2: The second largest eigenvalues of $\boldsymbol{\Theta}_{\mathrm{LP}}$ and $\boldsymbol{\Theta}_{\mathrm{TLP}}$ are given by

$$g_2(\boldsymbol{\Theta}_{\mathrm{LP}}) = 1 - \alpha\tilde{v}_{\hat{l}} \tag{34}$$

$$g_2(\boldsymbol{\Theta}_{\mathrm{TLP}}) = 1 - \tilde{v}_{\hat{l}}\phi_{\hat{l},\hat{l}-1,N_\mathrm{s}-1} \tag{35}$$

where $\hat{l} = N_\mathrm{c} - L + 1$.

Proof: See Section B in Appendix.

Based on Lemma 2, the rate of convergence depends on α in the case of LP and $\phi_{\hat{l},\hat{l}-1,N_\mathrm{s}-1}$ in the case of TLP. Furthermore, it can also be seen from Lemma 2 that the rate of convergence

in both cases also depends on the popularity of a particular content, i,e., the $(N_{\rm c}-L+1)$th content, or equivalently, the Lth most popular content.

Unlike RR and LRU, which do not rely on the content popularity information, prediction error in the content request probabilities could have an impact on either the STF or both the STF and the steady state of LP and TLP. Specifically, if there are errors in the prediction but the set of the L most popular contents is predicted correctly, then the predicted STF can differ from the actual STF but the steady state will not be affected. By contrast, if the predicted L most popular contents are different from the actual L most popular contents, then both the STF and the steady state from the prediction will differ from their respective actual values.

C. LRU

Using the definition of content-specific STF in eq. (11) and the state transition probability matrix of LRU in eq. (26), it can be shown that the mth element of the content-specific STF at $\boldsymbol{\eta}$ is given by:

$$u_{m,l,\mathrm{LRU}}(\boldsymbol{\eta})=\begin{cases}\displaystyle\sum_{\{k|m\in\mathcal{H}_{k,l}\}}\rho^{\mathrm{LRU}}_{e(k,m)|k}\eta_k, & \text{if } l\in\mathcal{C}_m,\\ -\eta_m, & \text{otherwise.}\end{cases} \tag{36}$$

Under the IRM model, the probabilities $\{\rho^{\mathrm{LRU}}_{e(k,m)|k}\}_{\forall k,\forall m\in\mathcal{H}_k}$ are constants and can be calculated. Given $\{\rho^{\mathrm{LRU}}_{e(k,m)|k}\}$, the following result regarding the steady state in the case of LRU can be found using the STF.

Theorem 2: The steady state $\boldsymbol{\eta}^\star$ in the case of LRU satisfies the following property:

$$\eta^\star_m\sum_{l\notin\mathcal{C}_m}\upsilon_l=\sum_{k\in\mathcal{H}_m}\upsilon_{e(m,k)}\rho^{\mathrm{LRU}}_{e(k,m)|k}\eta^\star_k. \tag{37}$$

Proof: See Section C in Appendix.

Comparing eq. (37) and eq. (29) reveals an interesting insight. Denote the steady state SCP

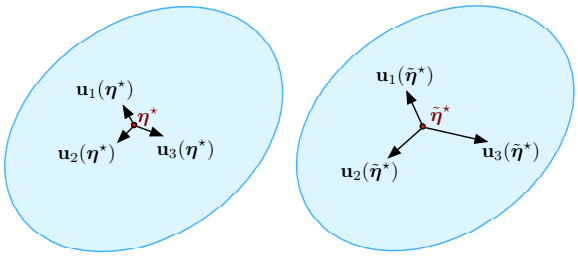
in the case of RR by $\boldsymbol{\eta}^{\star}_{\mathrm{RR}}$. The STF at the point $\boldsymbol{\eta}^{\star}_{\mathrm{RR}}$ in the case of LRU is given by:

$$
\begin{aligned}
&u_{m,\mathrm{LRU}}(\boldsymbol{\eta}^{\star}_{\mathrm{RR}}) \\
=&\sum_{l\in\mathcal{C}_m} \upsilon_l \cdot u_{m,l,\mathrm{LRU}}(\boldsymbol{\eta}^{\star}_{\mathrm{RR}}) + \sum_{l\notin\mathcal{C}_m} \upsilon_l \cdot u_{m,l,\mathrm{LRU}}(\boldsymbol{\eta}^{\star}_{\mathrm{RR}}) \\
=&\sum_{l\in\mathcal{C}_m} \upsilon_l \sum_{\{k|m\in\mathcal{H}_{k,l}\}} \rho^{\mathrm{LRU}}_{e(k,m)|k}\eta^{\star}_{k,\mathrm{RR}} - \sum_{l\notin\mathcal{C}_m} \upsilon_l \eta^{\star}_{m,\mathrm{RR}} \\
=&\sum_{l\in\mathcal{C}_m} \upsilon_l \sum_{\{k|m\in\mathcal{H}_{k,l}\}} \rho^{\mathrm{LRU}}_{e(k,m)|k}\eta^{\star}_{k,\mathrm{RR}} - \frac{1}{L}\sum_{k\in\mathcal{H}_m} \eta^{\star}_{k,\mathrm{RR}}\upsilon_{e(m,k)} \\
=&\sum_{l\in\mathcal{C}_m} \upsilon_l \sum_{\{k|m\in\mathcal{H}_{k,l}\}} \rho^{\mathrm{LRU}}_{e(k,m)|k}\eta^{\star}_{k,\mathrm{RR}} - \sum_{l\in\mathcal{C}_m} \upsilon_l \frac{1}{L}\sum_{\{k|m\in\mathcal{H}_{k,l}\}} \eta^{\star}_{k,\mathrm{RR}} \\
=&\sum_{l\in\mathcal{C}_m} \upsilon_l \sum_{\{k|m\in\mathcal{H}_{k,l}\}} \left(\rho^{\mathrm{LRU}}_{e(k,m)|k} - \frac{1}{L}\right)\eta^{\star}_{k,\mathrm{RR}},
\end{aligned} \tag{38}
$$

where the second step uses eq. (37) and the third step uses the property in eq. (29). The term $\rho^{\mathrm{LRU}}_{e(k,m)|k} - 1/L$ in eq. (38) is interesting as it shows the difference between the steady states in RR and LRU. Specifically, (38) shows that, compared to RR, the steady state of LRU favors states with popular contents.

As an example, consider the case when state m caches the L most popular contents. Then it follows that $\rho^{\mathrm{LRU}}_{e(k,m)|k} > 1/L$ in eq. (38) for any k such that $m \in \mathcal{H}_k$. This is true because content $e(k,m)$ is less popular than the other $L-1$ contents in state k, which are also cached by state m and therefore among the L most popular contents. Note that the constant $1/L$ can be considered as the probability that $e(k,m)$ is the LRU content when all cached contents have exactly the same request probability. As $\rho^{\mathrm{LRU}}_{e(k,m)|k} > 1/L$ for any k such that $m \in \mathcal{H}_k$ in eq. (38), $u_{m,\mathrm{LRU}}(\boldsymbol{\eta}^{\star}_{\mathrm{RR}}) > 0$, which shows that the STF of the LRU at $\boldsymbol{\eta}^{\star}_{\mathrm{RR}}$ points towards a direction that increases the probability of caching state m. Similarly, it can be shown that $u_{m',\mathrm{LRU}}(\boldsymbol{\eta}^{\star}_{\mathrm{RR}}) < 0$ if m' caches the least popular contents.

The above difference between the steady states of the RR and LRU roots from the difference in the information exploited in the two schemes. Unlike RR, which exploits no information and treats each cached content indifferently in every single replacement, the LRU exploits historical request information, which reflects the content popularity. As a result, LRU can converge to a steady state that caches popular contents with larger probabilities.



(a) Small $\|\mathbf{u}_l(\boldsymbol{\eta}^\star)\|$. (b) Large $\|\mathbf{u}_l(\tilde{\boldsymbol{\eta}}^\star)\|$.

Fig. 4: Illustration of decomposing the STF at the steady state.

VI. Discussions

In this section, we discuss the benefits of using the proposed STF to analyze replacement schemes in practice. First, we use an example to show how the STF can characterize the property of the stationary states. Then, we use another example to show how the STF can be used to compare the convergence rate of replacement schemes.

A. On the Steady State

Given two replacement schemes (or the same replacement scheme with different parameters), can we tell more about their steady states besides the cache hit probability?

At the steady state, the overall STF must be equal to $\mathbf{0}$ regardless of the replacement scheme. However, this does not mean that no replacement happens after the steady state is achieved. Instead, contents can still be evicted from or accepted into the cache, while the probabilities of the two events must be equal for any content at the steady state. Therefore, it is not difficult to see that, there can be more frequent replacements at the steady state of one replacement scheme than that of another. This frequency of replacement at a steady state can be analyzed by decomposing the STF into content-specific STF using eq. (12), as illustrated in Fig. 4. In the illustrated cases, we assume the same content request probabilities, while the content-specific STFs in Fig. 4a have much smaller norms than those in Fig. 4b. Correspondingly, there can be less frequent replacements at the steady state $\boldsymbol{\eta}^\star$ in Fig. 4a than at the steady state $\tilde{\boldsymbol{\eta}}^\star$ in Fig. 4b.

In the case when each replacement incurs a cost or when cache wear-out is a concern, characterizing the frequency of replacement can be of interest. Based on the above discussion, the weighted sum of the norm of content-specific STF can be used as a metric for comparing

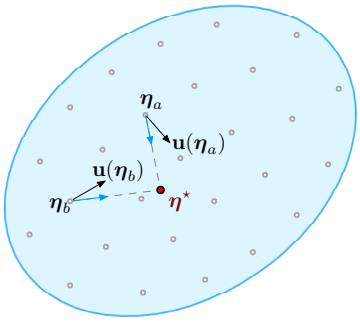


Fig. 5: Illustration of sampling the STF for characterizing the convergence rate.

the frequency of content replacement at the steady state of different replacement schemes. For example, a metric can be calculated as follows:

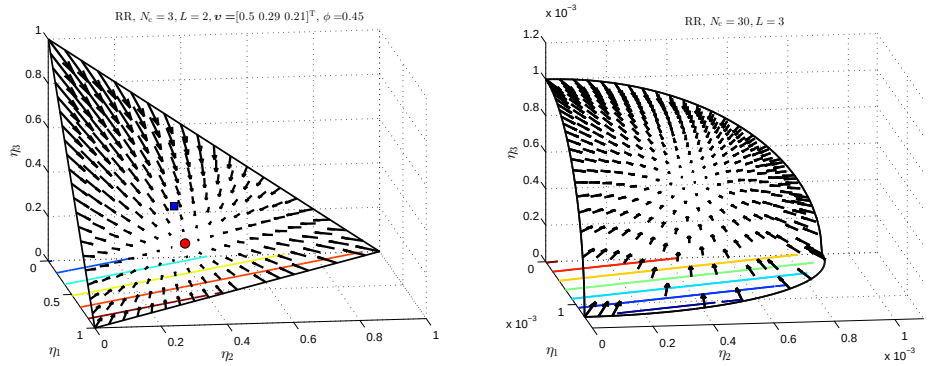
$$M(\boldsymbol{\eta}^\star) = \sum_{l \in \mathcal{C}} \upsilon_l \|\mathbf{u}_l(\boldsymbol{\eta}^\star)\|, \tag{39}$$

where the weights are the content request probabilities.

B. On the Convergence to the Steady State

We mentioned the rate of convergence and its relation with the second largest eigenvalue of the transition probability matrix $\boldsymbol{\Theta}$ in Section V. Since STF is a derivative of state transition probability matrix, it does not provide a new characterization of the rate of convergence in theory. However, we could use STF to develop a metric for comparing the convergence rate of different replacement schemes in practice.

For example, we can generate sample points in the state transition region, as illustrated using hollow circles in Fig. 5. Hypothetically, if the STF at every point of the state transition region points toward the steady state $\boldsymbol{\eta}^\star$, then the rate of convergence is determined by the strength (norm) of the STF. In practice, the STF at the sample points generally does not point straight toward the steady state. Nevertheless, we can project the STF at a sample point onto the connection line between that sample point and the steady state. This is illustrated with two example sample points, i.e., $\boldsymbol{\eta}_a$ and $\boldsymbol{\eta}_b$, in Fig. 5. In this figure, the solid circle filled with red represents the steady state $\boldsymbol{\eta}^\star$. The black arrows at sample points $\boldsymbol{\eta}_a$ and $\boldsymbol{\eta}_b$ represent the STF $\mathbf{u}(\boldsymbol{\eta}_a)$ and $\mathbf{u}(\boldsymbol{\eta}_b)$, respectively. The two dashed lines connect $\boldsymbol{\eta}_a$ and $\boldsymbol{\eta}_b$ with the steady state $\boldsymbol{\eta}^\star$. The two blue arrows on the dashed lines represent the projection of $\mathbf{u}(\boldsymbol{\eta}_a)$ and $\mathbf{u}(\boldsymbol{\eta}_b)$, respectively. The norm of the projection, aggregated over all sample points, can provide a metric



(a) STF of RR, $\phi = 0.45, \boldsymbol{\upsilon} = [0.5, 0.29, 0.21]^{\mathrm{T}}$. (b) STF of RR, $N_{\mathrm{c}} = 30, L = 3$, in a 3-D subspace.

Fig. 6: STF of RR in 3-D.

for characterizing the rate of convergence of replacement schemes. The accuracy of this approach depends on the number and locations of the chosen sample points.

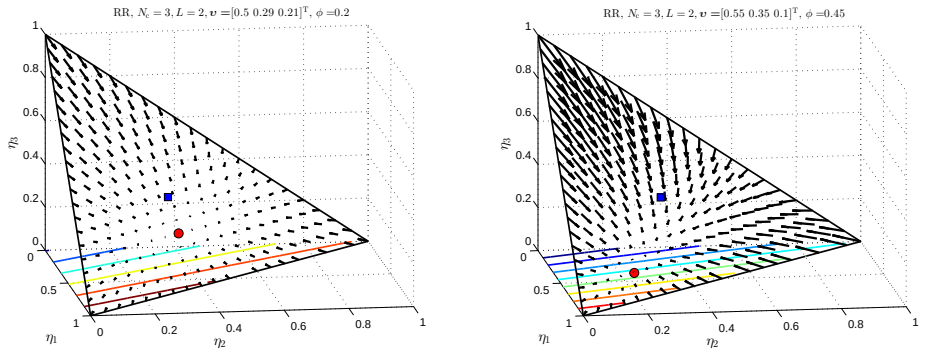
VII. Numerical Results

The numerical examples are organized into three sections. The first section demonstrates STFs obtained from analysis. The second section demonstrates STFs obtained from simulations and compare it with the analytical results. The third section demonstrates and compares the CCP and cache hit probability of the considered schemes to reveal the impact of different STFs.

A. STF - Analytical

In this section, the analytical STFs of RR, LP, TLP, and LRU are demonstrated. In general, STF can be of high dimensions. We limit most of our demonstration to the case of three dimensions, as three-dimensional fields can be very well visualized and illustrated. A three-dimensional subspace in a high-dimensional STF is also illustrated.

Fig. 6a demonstrates a three-dimensional STF of RR. In this figure, $N_{\mathrm{c}} = 3$, $L = 2$, and therefore there are only three cache states (i.e., $\mathcal{C}_1 = \{1, 2\}, \mathcal{C}_2 = \{1, 3\}, \mathcal{C}_3 = \{2, 3\}$). The x, y, and z axes correspond to the SCP for the cache states 1, 2, and 3, respectively. The triangular area is the state transition domain $\mathcal{D}$, the square marker represents the center of the triangle, and the circle represents the steady-state SCP $\boldsymbol{\eta}^{\star}$ in this example. The STF at a point in $\mathcal{D}$ is represented by an arrow originating from that point, while the strength and direction of the



(a) STF of RR, $\phi = 0.2, \boldsymbol{\upsilon} = [0.5, 0.29, 0.21]^{\mathrm{T}}$. (b) STF of RR, $\phi = 0.45, \boldsymbol{\upsilon} = [0.55, 0.35, 0.1]^{\mathrm{T}}$.

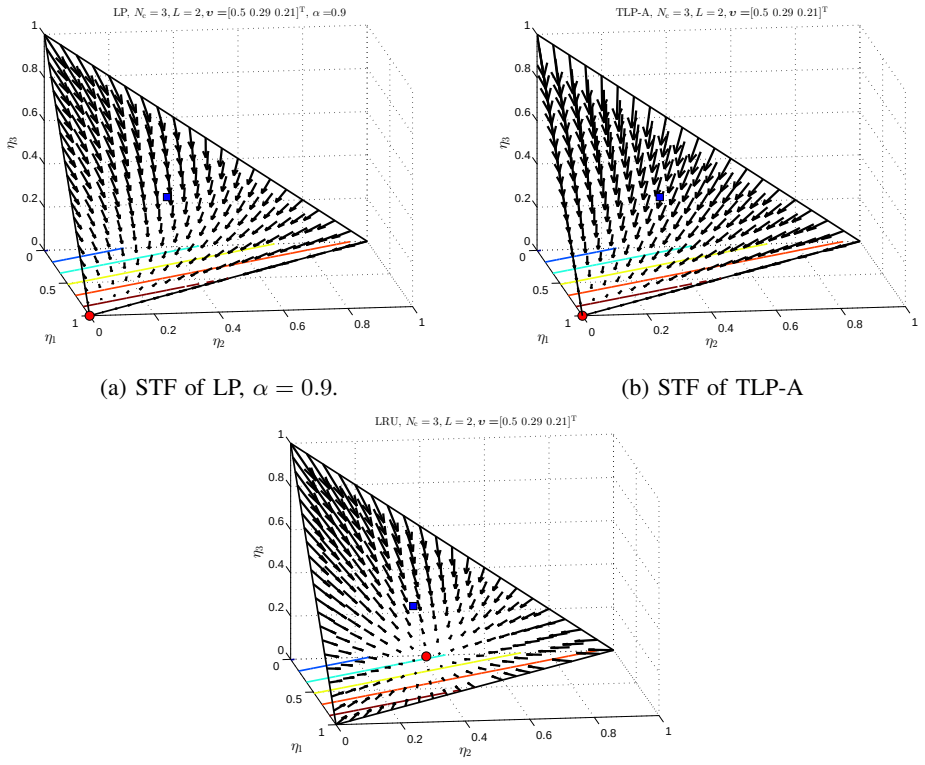
Fig. 7: The impact of $\boldsymbol{\upsilon}$ and ϕ on the STF of RR.

STF are shown by the length of the arrow and the direction of the arrowhead, respectively. The straight lines in the x-y plane show the contour of the cache hit probability for the SCP.

Fig. 6b demonstrates part of a high-dimensional STF over the surface of an ellipsoid in a three-dimensional subspace. In this example, $N_{\mathrm{c}} = 30$, $L = 3$, and there are 4060 cache states. Three mutually-neighbor cache states are selected, corresponding to the three-dimensional subspace in the figure. The STF over the surface of an ellipsoid in this subspace is demonstrated as an example. The x, y, and z axes correspond to the SCP for the three selected cache states. Unlike the case in Fig. 6a, the SCPs in 6b are small and do not sum up to 1 since there are many other states. Fig. 6b serves as an example of high-dimensional STF.

Fig. 7 demonstrates the impact of the content popularity $\boldsymbol{\upsilon}$ and the parameter ϕ on the STF of RR. Fig. 7a shows the STF under the same settings as in Fig. 6a except that ϕ is decreased from 0.45 to 0.2. Two observations can be made by comparing Fig. 7a with Fig. 6a. First, the steady-state SCP in both cases are identical, which confirms Theorem 1. Second, the strength of STF at any given point in Fig. 7a is weaker as compared to that in Fig. 6a, which implies a longer mixing time. Fig. 7b shows the STF under the same settings as in Fig. 6a except a change in the content popularity $\boldsymbol{\upsilon}$. It can be seen from this figure that the steady state also changes following the change in content popularity. Comparing Fig. 7b with Fig. 6a, the impact of content popularity on the STF can be observed.

Fig. 8 demonstrates three-dimensional STFs of LP, TLP, and LRU. In all three plots in Fig. 8, $\boldsymbol{\upsilon}$ is set to $[0.5, 0.29, 0.21]^{\mathrm{T}}$. In Figs. 8a and 8b, the steady state is the vertex of the triangle with the highest cache hit probability. The difference is that the STF in Fig. 8a lead to a 'curvy'



(a) STF of LP, $\alpha = 0.9$.

(b) STF of TLP-A

(c) STF of LRU

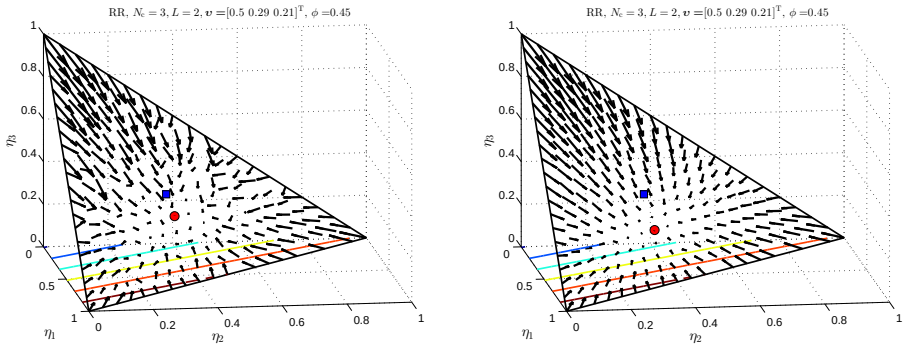
Fig. 8: The STF of LP, TLP, and LRU in 3-D.

path towards the steady state in Fig. 8a while the curvature of paths in Fig. 8b is much smaller. This reflects the fact that TLP makes replacements along the path which increases the cache hit probability most rapidly, bearing a certain resemblance to the 'steepest ascent' in gradient ascent. Fig. 8c appears similar to Fig. 6a. However, it can be observed that, compared to RR, the steady state of LRU assigns a larger caching probability to states with more popular contents. This is consistent with eq. (38) and the fact that RR exploits no historical information while making cache replacements.

B. STF - Numerical

In this section, we demonstrate, using RR and LRU as examples, STFs obtained from simulations and compare them with the analytical STF from the preceding section.

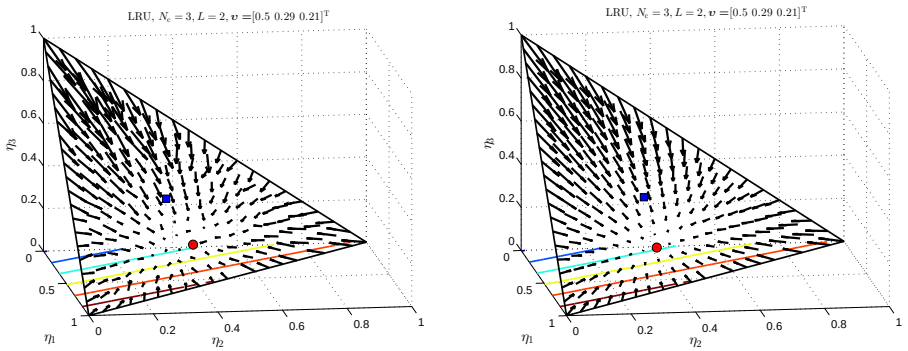
Fig. 9 shows the STF of RR generated from simulations. The settings on ϕ and $\boldsymbol{v}$ in Fig. 9 are exactly the same as those in Fig. 6a. For each point in the STF, M realizations of states are generated based on the corresponding SCP. For each realization, R content requests are generated



(a) STF of RR from simulation, $M = 100$, $R = 100$.

(b) STF of RR from simulation, $M = 1000$, $R = 1000$.

Fig. 9: The STF of RR from simulations.



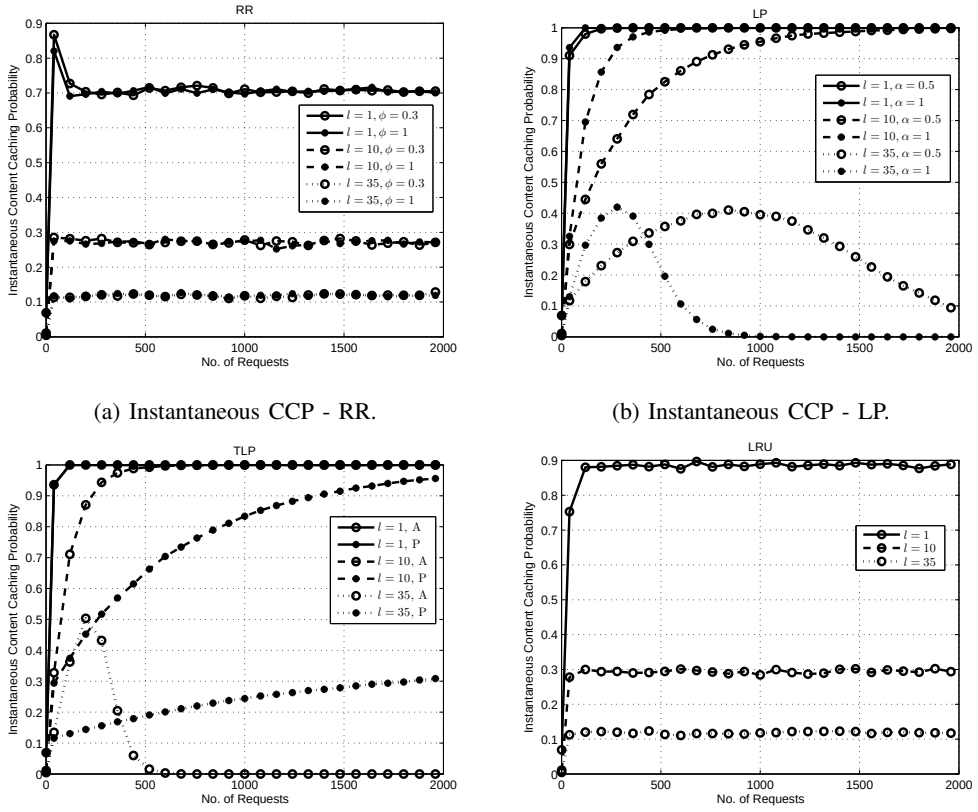
(a) STF of LRU from simulation, $R = 500$.

(b) STF of LRU from simulation, $R = 10000$.

Fig. 10: The STF of LRU from simulations.

based on the content popularity. Each data point (i.e., each arrow) in Figs. 9a and 9b is obtained from averaging the state transitions following the $M \times R$ requests. In Fig. 9a, M and R are both set to 100. It can be seen that the STF is not accurate, especially in the area close to the steady state, due to insufficient samples. In addition, the arrows point to a steady state slightly deviated from the true steady state in Fig. 6a. In Fig. 9b, M and R are both increased to 1000. It can be seen that the resulting STF generated based on simulation in Fig. 9b becomes an exact match for the analytical STF in Fig. 6a.

Fig. 10 shows the STF of LRU generated from simulations. The settings on $\boldsymbol{v}$ in Fig. 10 is exactly the same as that in Fig. 8c. Since LRU depends on request history, the simulation method used for Fig. 9 based on randomly generated states cannot be applied. Instead, for each point in the STF, R content requests are generated based on the content popularity. The STF is



(a) Instantaneous CCP - RR.

(b) Instantaneous CCP - LP.

(c) Instantaneous CCP - TLP.

(d) Instantaneous CCP - LRU.

Fig. 11: Demonstration of instantaneous CCP of the replacement schemes, $N_{\mathrm{c}} = 1000, L = 30$.

generated based on the state transitions following the R requests. Figs. 10a and 10b demonstrate a similar result as that from Figs. 9a and 9b: the STF from simulations can deviate from the analytical STF when the number of samples is small, while the two become an almost exact match when the sampled number of requests is sufficiently large.

C. Instantaneous CCP

This section demonstrates the instantaneous CCP of RR, LP, TLP, and LRU, and relate the results to the STF demonstrated in the preceding sections.

The number of contents N_{c} and the cache size L are set to 1000 and 30, respectively. For each of the four considered replacement schemes, the simulation consists of 5000 rounds. For each round, 2000 content requests are generated randomly based on a Zipf's distribution with parameter 0.8. The contents are sorted based on the request probability in a decreasing order.

The cache is empty at the beginning. The instantaneous CCP for each content after each request is obtained and averaged over the 5000 rounds.

The resulting CCP for three selected contents, i.e., contents 1, 10, and 35, are shown in Fig. 11. It can be seen from Figs. 11a and 11d that, starting with an empty cache, RR and LRU becomes stationary faster than LP and TLP, which are shown in Fig. 11b and Fig. 11c, respectively. In addition, by comparing Figs. 11a and 11d, it can be seen that LRU caches popular contents, e.g., content 1, with larger probabilities than RR. This is consistent with the observation from comparing Fig. 8c and Fig. 6a. The impact of α on the performance of LP can be seen from Fig. 11b, while the difference between TLP-A and TLP-P can be seen from Fig. 11c. In the cases of LP and TLP, the cache hit probability of content 35 first increases and then decreases to zero. This corresponds to the 'curvy' paths in the STF as shown in Fig. 8a and Fig. 8b.

VIII. Conclusion

We have revisited the problem of modeling and analyzing cache replacement schemes under IRM with the objective of providing a rigorous yet intuitive general model from a novel perspective. Through this work, we have developed a basic tool set based on STF to characterize and illustrate cache replacement schemes. Our investigation has also been targeted at revealing insights regarding the relation between content popularity, knowledge of content popularity exploited by replacement schemes, and the resulting STFs. The model and methodology we have established in this paper can also be applied to multi-level cache and cache networks after appropriate extensions.

Appendix

A. Proof of Theorem 1

We first prove, using the STF, that the steady state is independent on ϕ. It can be seen from eq. (28) that ϕ is just a scaling factor in $u_{m,l,\mathrm{RR}}$. Moreover, the scaling factor is the same for any state m and content l. Therefore, we can define a base STF such that at point $\boldsymbol{\eta}$ it satisfies:

$$\bar{u}_{m,l,\mathrm{RR}}(\boldsymbol{\eta}) = \begin{cases} \sum\limits_{\{k|m\in\mathcal{H}_{k,l}\}} \eta_k, & \text{if } l \in \mathcal{C}_m \\ -L\eta_m, & \text{otherwise.} \end{cases} \tag{40}$$

Then, it is easy to show that:

$$\mathbf{u}_{l,\mathrm{RR}}(\boldsymbol{\eta}) = \phi\bar{\mathbf{u}}_{l,\mathrm{RR}}(\boldsymbol{\eta}), \tag{41a}$$

$$\mathbf{u}_{\mathrm{RR}}(\boldsymbol{\eta}) = \phi\bar{\mathbf{u}}_{\mathrm{RR}}(\boldsymbol{\eta}). \tag{41b}$$

Accordingly, a change in ϕ can change the strength of the STF but does not alter the direction of the STF at any point in the state transition domain. Therefore, the steady state of RR must be independent on ϕ.

Next, we prove the property of the steady state. Based on the definition of STF in eq. (9), the STF at the steady state SCP $\boldsymbol{\eta}^\star$ must be equal to $\mathbf{0}$. It follows that:

$$\begin{aligned} u_{m,\mathrm{RR}}(\boldsymbol{\eta}^\star) &= \sum_{l\in\mathcal{C}_m} \upsilon_l \cdot u_{m,l,\mathrm{RR}}(\boldsymbol{\eta}^\star) + \sum_{l\notin\mathcal{C}_m} \upsilon_l \cdot u_{m,l,\mathrm{RR}}(\boldsymbol{\eta}^\star) \\ &= \sum_{l\in\mathcal{C}_m} \upsilon_l \phi \sum_{\{k|m\in\mathcal{H}_{k,l}\}} \eta_k^\star + \sum_{l\notin\mathcal{C}_m} \upsilon_l(-L\phi)\eta_m^\star \\ &= 0, \end{aligned} \tag{42}$$

which must hold for any $m \in \mathcal{S}$. Based on the definition of neighbors and content-specific neighbors, it can be seen that:

$$\sum_{l\in\mathcal{C}_m} \upsilon_l \sum_{\{k|m\in\mathcal{H}_{k,l}\}} \eta_k^\star = \sum_{k\in\mathcal{H}_m} \upsilon_{e(m,k)}\eta_k^\star. \tag{43}$$

Combining eq. (43) and eq. (42) gives eq. (29). ■

B. Proof of Lemma 2

First, we will prove that the second largest eigenvalue of both $\boldsymbol{\Theta}_{\mathrm{LP}}$ and $\boldsymbol{\Theta}_{\mathrm{TLP}}$ is the $(N_\mathrm{s}-1, N_\mathrm{s}-1)$th element. In the case of LP, the sum probability of state k transitioning into any other state is given by $\alpha\sum_{l\in\mathcal{C}_{\tilde{k}\uparrow}}\upsilon_l$, which is non-increasing with k. Accordingly, $\boldsymbol{\Theta}_{\mathrm{LP}}(m,m) \geq \boldsymbol{\Theta}_{\mathrm{LP}}(k,k)$ if $m > k$. Similarly, the same result can be shown for the TLP.

Second, as the states are sorted based on the sum predicted request probability of their cached contents, it can be seen that $e(N_\mathrm{s}, N_\mathrm{s}-1)$ is the $(N_\mathrm{c}-L+1)$th content. Based on eq. (21), it can be seen that $\boldsymbol{\Theta}_{\mathrm{LP}}(N_\mathrm{s}-1, N_\mathrm{s}-1)$ is equal to $1-\alpha\tilde{\upsilon}_{\hat{l}}$ with $\hat{l}$ denoting $N_\mathrm{c}-L+1$. Similarly, $\boldsymbol{\Theta}_{\mathrm{TLP}}(N_\mathrm{s}-1, N_\mathrm{s}-1)$ is equal to $1-\tilde{\upsilon}_{\hat{l}}\phi_{\hat{l},\hat{l}-1,N_\mathrm{s}-1}$ based on eq. (25). ■

C. Proof of Theorem 2

The STF at the steady state SCP $\boldsymbol{\eta}^\star$ must be equal to $\mathbf{0}$. It follows that:

$$\begin{aligned} u_{m,\mathrm{LRU}}(\boldsymbol{\eta}^\star) &= \sum_{l\in\mathcal{C}_m} \upsilon_l \cdot u_{m,l,\mathrm{LRU}}(\boldsymbol{\eta}^\star) + \sum_{l\notin\mathcal{C}_m} \upsilon_l \cdot u_{m,l,\mathrm{LRU}}(\boldsymbol{\eta}^\star) \\ &= \sum_{l\in\mathcal{C}_m} \upsilon_l \sum_{\{k|m\in\mathcal{H}_{k,l}\}} \rho^{\mathrm{LRU}}_{e(k,m)|k}\eta_k^\star - \sum_{l\notin\mathcal{C}_m} \upsilon_l\eta_m^\star \\ &= 0, \end{aligned} \tag{44}$$

which must hold for any $m \in \mathcal{S}$. It can be shown that:

$$\sum_{l \in \mathcal{C}_m} \upsilon_l \sum_{\{k | m \in \mathcal{H}_{k,l}\}} \rho^{\text{LRU}}_{e(k,m)|k} \eta^\star_k = \sum_{k \in \mathcal{H}_m} \upsilon_{e(m,k)} \rho^{\text{LRU}}_{e(k,m)|k} \eta^\star_k. \tag{45}$$

Combining eq. (45) and eq. (44) gives eq. (37). ■

REFERENCES

[1] Z. Piao, M. Peng, Y. Liu, and M. Daneshmand, "Recent Advances of Edge Cache in Radio Access Networks for Internet of Things: Techniques, Performances, and Challenges," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1010–1028, Feb. 2019.

[2] E. K. Markakis, K. Karras, A. Sideris, G. Alexiou, and E. Pallis, "Computing, Caching, and Communication at the Edge: The Cornerstone for Building a Versatile 5G Ecosystem," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 152–157, Nov. 2017.

[3] M. Tang, L. Gao, and J. Huang, "Enabling Edge Cooperation in Tactile Internet via 3C Resource Sharing," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2444–2454, Nov. 2018.

[4] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao, and X. Shen, "Cooperative Edge Caching in User-Centric Clustered Mobile Networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 8, pp. 1791–1805, Aug. 2018.

[5] M. Emara, H. Elsawy, S. Sorour, S. Al-Ghadhban, M. Alouini, and T. Y. Al-Naffouri, "Optimal Caching in 5G Networks With Opportunistic Spectrum Access," *IEEE Trans. Wireless Commun.*, vol. 17, no. 7, pp. 4447–4461, July 2018.

[6] T. X. Vu, S. Chatzinotas, B. Ottersten, and T. Q. Duong, "Energy Minimization for Cache-Assisted Content Delivery Networks With Wireless Backhaul," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 332–335, June 2018.

[7] G. Lee, I. Jang, S. Pack, and X. Shen, "FW-DAS: Fast Wireless Data Access Scheme in Mobile Networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 8, pp. 4260–4272, Aug. 2014.

[8] E. Bastug, M. Bennis, and M. Debbah, "Living on the Edge: The Role of Proactive Caching in 5G Wireless Networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

[9] K. N. Doan, T. Van Nguyen, T. Q. S. Quek, and H. Shin, "Content-Aware Proactive Caching for Backhaul Offloading in Cellular Network," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3128–3140, May 2018.

[10] J. Gao, L. Zhao, and L. Sun, "Probabilistic Caching as Mixed Strategies in Spatially-Coupled Edge Caching," in *Proc. 29th Biennial Symp. Commun.*, Toronto, Canada, 2018.

[11] J. Qiao, Y. He, and X. Shen, "Proactive Caching for Mobile Video Streaming in Millimeter Wave 5G Networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 7187–7198, Oct. 2016.

[12] S. O. Somuyiwa, A. György, and D. Gündüz, "A Reinforcement-Learning Approach to Proactive Caching in Wireless Networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1331–1344, June 2018.

[13] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online Coded Caching," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 836–845, Apr. 2016.

[14] S. Podlipnig and L. Böszörmenyi, "A Survey of Web Cache Replacement Strategies," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 374–398, Dec. 2003.

[15] L. A. Belady, "A Study of Replacement Algorithms for a Virtual-Storage Computer," *IBM Sys. J.*, vol. 5, no. 2, pp. 78–101, 1966.

[16] D. D. Sleator and R. E. Tarjan, "Amortized Efficiency of List Update and Paging Rules," *Commun. ACM*, vol. 28 , no. 2, pp. 202–208, Feb. 1985.

[17] G. S. Rao, "Performance Analysis of Cache Memories," *J. ACM*, vol. 25, no. 3, pp. 378-395, July 1978.

[18] A. R. Karlin,, S. J. Phillips, and P. Raghavan, "Markov Paging," *SIAM J. Comput.*, vol. 30, no. 3, pp. 906-922, Aug. 2000.

[19] R. Hirade and T. Osogami. “Analysis of Page Replacement Policies in the Fluid Limit,” *Operations Research*, vol. 58, no. 4, pp. 971-984, July 2010.

[20] H. Gomaa, G. G. Messier, C. Williamson, and R. Davies, “Estimating Instantaneous Cache Hit Ratio Using Markov Chain Analysis,” *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1472–1483, Oct. 2013.

[21] S. Tarnoi, V. Suppakitpaisarn, W. Kumwilaisak, and Y. Ji, “Performance Analysis of Probabilistic Caching Scheme using Markov Chains,” in *Proc. IEEE LCN*, Clearwater Beach, USA, 2015, pp. 46–54.

[22] J. Li, S. Shakkottai, J. C. S. Lui, and V. Subramanian, “Accurate Learning or Fast Mixing? Dynamic Adaptability of Caching Algorithms,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1314–1330, June 2018.

[23] J. Gao, S. Zhang, L. Zhao, and X. Shen, “The Design of Dynamic Probabilistic Caching with Time-Varying Content Popularity,” submitted to *IEEE Trans. Mobile Comput.*, under review.

[24] L. Chang, J. Pan, and M. Xing, “Effective Utilization of User Resources in PA-VoD Systems with Channel Heterogeneity,” *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 227–236, Sept. 2013.

[25] M. Fiore, C. Casetti, and C. Chiasserini, “Caching Strategies Based on Information Density Estimation in Wireless Ad Hoc Networks,” *IEEE Trans. Veh. Technol.*, vol. 60, no. 5, pp. 2194–2208, June 2011.

[26] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, K. Drira, and H. Mathkour, “Least Fresh First Cache Replacement Policy for NDN-based IoT networks,” *Pervasive Mob. Comput.*, vol 52, pp. 60–70, Jan. 2019.

[27] Z. H. Meybodi, J. Abouei, and A. H. F. Raouf, “Cache Replacement Schemes based on Adaptive Time Window for Video on Demand Services in Femtocell Networks,” *IEEE Trans. on Mobile Comput.*, vol. 18, no. 7, pp. 1476–1487, July 2019.

[28] N. Kamiyama, Y. Nakano, and K. Shiomoto, “Cache Replacement Based on Distance to Origin Servers,” *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 4, pp. 848–859, Dec. 2016.

[29] A. Chattopadhyay, B. Błaszczyszyn, and H. P. Keeler, “Gibbsian On-Line Distributed Content Caching Strategy for Cellular Networks,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 2, pp. 969–981, Feb. 2018.

[30] E. Leonardi and G. Neglia, “Implicit Coordination of Caches in Small Cell Networks Under Unknown Popularity Profiles,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1276–1285, June 2018.

[31] J. Gao, L. Zhao, and X. Shen, “The Study of Caching via State Transition Field - the Case of Time-Varying Popularity,” *IEEE Trans. Wireless Commun.*, accepted.

[32] G. S. Paschos, G. Iosifidis, M. Tao, D. Towsley, and G. Caire, “The Role of Caching in Future Communication Systems and Networks,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1111–1125, June 2018.

[33] D. Levin, Y. Peres, E. Wilmer, *Markov Chains and Mixing Times*. American Mathematical Society, Providence, RI, USA, 2008.

[34] S. Arora. *Random walks, Markov Chains, and How to Analyse Them*. Lecture Notes, Department of Computer Science, Princeton University, 2013.

[35] S. G. Walker, “Bounds for the Second Largest Eigenvalue of a Transition Matrix,” *Linear and Multilinear Algebra*, vol. 59, no. 7, pp. 755-760, Apr. 2011.