Network Cache Design under Stationary Requests: Exact Analysis and Poisson Approximation

Nitish K. Panigrahy, Jian Li, Don Towsley
College of Information and Computer Sciences
University of Massachusetts Amherst
{nitish, jianli, towsley}@cs.umass.edu

***Abstract*—The design of caching algorithms to maximize hit probability has been extensively studied. In this paper, we associate each content with a utility, which is a function of either corresponding content hit rate or hit probability. We formulate a cache optimization problem to maximize the sum of utilities over all contents under stationary and ergodic request process. This problem is non-convex in general but we reformulate it as a convex optimization problem when the inter-request time (irt) distribution has a non-increasing hazard rate function. We provide explicit optimal solutions for some irt distributions, and compare the solutions of the hit-rate based (HRB) and hit-probability based (HPB) problems. We also propose decentralized algorithms that can be implemented using limited information and are guaranteed to provide optimal solutions. We find that decentralized algorithms that solve HRB are more robust than decentralized HPB algorithms. Informed by these results, we further propose lightweight Poisson approximate decentralized and online algorithms that are accurate and efficient in achieving optimal hit rates and hit probabilities.**

I. Introduction

Caches play a prominent role in networks and distributed systems for improving system performance. Since the number of contents in a system is typically significantly larger than cache capacity, the design of a caching algorithm typically focuses on maximizing the number of requests that can be served from the cache. Considerable research has focused on the analysis of caching algorithms using the metric of hit probability under the Independent Reference Model (IRM) [1], [7], [20], [6], [27], [24], [29]. However, *hit rate* [13] is a more relevant performance metric in real systems. For example, pricing based on hit rate is preferable to that based on cache occupancy from the perspective of a service provider [25]. Furthermore, one goal of a service provider in designing hierarchical caches would be to minimize the internal bandwidth cost, which can be characterized with a utility function $U_i = -C_i(m_i)$, where $C_i(m_i)$ is the cost associated with miss rate m_i for content i. Therefore, we focus on the metric hit rate in this paper.

Recently there has been a tremendous increase in the demand for different types of content with different quality of service requirements; consequently, user needs have become more heterogeneous. In order to meet such challenges, content delivery networks need to incorporate service differentiation among different classes of contents and applications. Though considerable literature has focused on the design of fair and efficient caching algorithms for content distribution, little work has focused on the provision of multiple levels of service in network and web caches.

Moreover, cache behaviors of different contents are strongly coupled by conventional caching algorithms such as LRU [1], [24], [16], which make it difficult for cache service providers to provide differential services. In this paper, we focus on Time-to-Live (TTL) caches. When a content is inserted into the cache due to a cache miss, a timer is set. Timer value can differ for different contents. All requests to a content before the expiration of its timer results in a cache hit, and the first request after the expiration of its timer yields a cache miss. This ability to decouple the behaviors of different contents make the TTL policy an interesting alternative to more popular algorithms like LRU. Moreover, the TTL policy has been shown to mimic the behaviors of many caching algorithms.

In this paper, we consider a utility-driven caching framework, where each content is associated with a utility. Content is stored and managed in the cache so as to maximize the aggregate utility for all content. A related problem has been considered in [7], where the authors formulated a *Hit-probability Based Cache Utility Maximization* (HPB-CUM) framework under IRM. The objective is to maximize the sum of utilities under a cache capacity constraint when utilities are increasing, continuously differentiable, and strictly concave function of hit probability. [7], [14] and [29] characterized optimal TTL cache policies, and also proposed distributed cache management algorithms. Here, we focus on utilities as functions of hit rates.

While characterization of hit rate under IRM is valuable, real-world request processes exhibit changes in popularity and temporal correlations in requests [35], [4]. To account for them, in this paper, *we consider a very general traffic model where requests for distinct contents are described by mutually independent stationary and ergodic point processes* [2].

A. Contributions

Our main contributions in this paper can be summarized as follows.

1) We formulate a *Hit-rate Based Cache Utility Maximization* (HRB-CUM) framework for maximizing aggregate content utilities subject to an expected cache size constraint at the service provider. In general, *HRB-CUM with TTL caches under general stationary request process is a non-convex optimization problem.* We develop a convex optimization problem

for the case that the inter-request time distributions have non-increasing hazard rate. This is an important case since inter-request times are often highly variable.

2) We compare hit rate based approaches to hit probability based approaches when utilities come from a family of β-fair utility functions. We find that HRB-CUM and HPB-CUM are identical under log utility function with $\beta = 1$. However, for $\beta < 1$, there exists a threshold such that HRB-CUM favors more popular contents over HPB-CUM, i.e., popular contents will be cached under HRB-CUM. The reverse behavior holds for $\beta > 1$.

3) We propose decentralized algorithms that adapts to different stationary requests using limited information. We prove that all our decentralized algorithms obtain the optimal solutions. We find that the corresponding decentralized algorithms for HRB-CUM are more robust and stable than those for HPB-CUM with respect to (w.r.t.) convergence rate.

4) Inspired by the analysis of decentralized algorithms, we further propose a lightweight Poisson approximate online algorithm where we apply the dual designed for the case of requests described by a Poisson process to a workload where requests are described by stationary request processes. Such a solution does not involve solving any non-linear equations and hence is computationally efficient.

In particular, we consider the 2-MMPP and allow it to characterize its limiting behavior in terms of state transition rates. We find that when the transition rates both go to infinity, 2-MMPP is equivalent to a Poisson process, i.e., and our Poisson approximation is exact. We numerically show that our approximation is accurate in achieving near optimal hit rates and hit probabilities.

This analysis provides significant insights in modeling real traffic with Poisson process and also verify the robustness and wide applicability of Poisson process.

B. Related Work and Organization

Network Utility Maximization: Utility functions have been widely used in the performance analysis of computer networks, which define different fairness. Since Kelly's seminal work [22], [23], a rich literature uses network utility maximization problem in the analysis of throughput maximization, dynamic allocation, network routing etc and we do not attempt to provide a detailed overview here.

Time-To-Live Caches: TTL caches have been employed in the Domain Name System (DNS) since the early days of Internet [20]. More recently, it has gained attention due to the case that it can easily be analyzed and can be used to model the behaviors of caching algorithms such as LRU. The TTL cache has been shown to provide accurate estimates of the performance of large caches, as first introduced for LRU under IRM [8], [6] through the notion of *cache characteristic time*. It has been further generalized to other settings [3], [13], [17], [16]. The accuracy of the TTL cache is theoretically justified under IRM [3] and stationary processes [18], and numerically verified under renewal processes [16]. A recent paper [10] has tackled a similar problem close to ours, which focuses on maximizing hit probabilities under DHR demands. Instead, we focus on optimizing the total utilities of cache contents.

The paper is organized as follows. The next section contains some technical preliminaries. We formulate the HRB-CUM and HPB-CUM under general stationary requests in Section III, and present some specific inter-request processes under which HRB-CUM and HPB-CUM become convex optimization problems in Section IV. We compare their performance both theoretically and numerically in Section V. We develop decentralized algorithms and give its performance evaluations in Section VI. We present Poisson approximate online algorithms in Section VII. We conclude the paper in Section VIII.

II. Technical Preliminaries

We consider a cache of size B serving n distinct contents.

A. Content Request Process

In this paper, the request processes for distinct contents are described by mutually independent stationary and ergodic simple point process as [2], [18]. Our model generalizes the simplest and widely used Independence Reference Model (IRM) [1], where requests are described by Poisson processes.

Let $\{t_{ik}, k \in \mathbb{Z}\}$ represent successive request times to content $i = 1, \cdots, n$. Let $X_{ik} = t_{ik} - t_{i(k-1)}$ denote the inter-request times for a particular content i. We consider $\{X_{ik}\}_{k\geq 1}$ to be a stationary point process with cumulative inter-request time (*irt*) distribution functions (c.d.f.) satisfying [2]

$$F_i(t) = \mathbb{P}(X_{ik} \leq t),\ i = 1, \cdots, n. \tag{1}$$

For example, for the l-state MMPP, F is a mixture of l exponential distributions.

The mean request rate μ_i for content i is then given by

$$\mu_i = \frac{1}{\mathbb{E}[X_{ik}]} = \frac{1}{\int_0^\infty (1 - F_i(t))dt}. \tag{2}$$

Denote by $\hat{F}_i(t)$ the c.d.f. of the age associated with the irt distribution for content i, satisfying [2]

$$\hat{F}_i(t) = \mu_i \int_0^t (1 - F(x))dx. \tag{3}$$

It is known that [2] the popularity (requested probability) of content i satisfies

$$p_i = \mu_i/\mu, \tag{4}$$

with $\mu = \sum_{i=1}^n \mu_i$.

In our work, we consider various irt distributions, including exponential, Pareto, hyperexponential and MMPP.

B. Content Popularity

Whereas our analytical results hold for any popularity law, in our numerical studies we will use the Zipf distribution as this distribution is widely used in cache studies, and has been frequently observed in real traffic measurements [5]. Under the Zipf distribution, the probability of requesting the i-th most popular content is A/i^α, where α is the Zipf parameter depending on the application [15], and A is the normalization factor satisfying $\sum_{i=1}^n p_i = 1$.

C. TTL Caches

In a TTL cache, each content i is associated with a timer t_i. When content i is requested, there are two cases: (i) if the content is not in the cache (miss), then content i is inserted into the cache and its timer is set to t_i; (ii) if the content is in the cache (hit), then the timer associated with content i is reset. The timer decreases at a constant rate and the content is evicted once its timer expires. This is referred to as a *Reset TTL Cache*. We can control the hit probability of each content by adjusting its timer value.

Denote the *hit rate* and *hit probability* of content i as λ_i and h_i, respectively, then from the analysis of previous work [12], the hit probability and hit rate for a reset TTL cache can be computed as

$$h_i = F_i(t_i), \quad \lambda_i = \mu_i F_i(t_i), \tag{5}$$

respectively, where requests for content i follow a request process as described in Section II-A.

Let h_i^{in} be the *time-average probability* that content i is in the cache (i.e., *occupancy probability*), then we have [16], [10]

$$h_i^{\text{in}} = \hat{F}_i(t_i). \tag{6}$$

In particular, our model reduces to classical IRM when the inter-request time are exponentially distributed, i.e., Poisson arrival process [7], with $F_i(t_i) = 1 - e^{-\mu_i t_i}$ and $h_i = h_i^{\text{in}}$, based on PASTA property [26].

D. Utility Function and Fairness

Utility functions capture the satisfaction perceived by a content provider. Here, we focus on the widely used β-fair utility functions [33] given by

$$U_i(x) = \begin{cases} w_i \frac{x^{1-\beta}}{1-\beta}, & \beta \geq 0, \beta \neq 1; \\ w_i \log x, & \beta = 1, \end{cases} \tag{7}$$

where $w_i > 0$ denotes a weight associated with content i.

III. Cache Utility Maximization

In this section, we formulate a utility maximization problem for cache management (CUM). In particular, we consider a formulation based on hit rate (HRB-CUM)[1]. As mentioned in the introduction, one can also formulate a problem based on hit probability. A similar formulation for HPB-CUM is available in Appendix X-A.

We are interested in optimizing the sum of utilities over all contents,

$$\max_{\{t_1,\cdots,t_n\}} \quad \sum_{i=1}^{n} U_i(\lambda_i^r(t_i)) \tag{8a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} h_i^{r,\text{in}}(t_i) \leq B, \tag{8b}$$

$$0 \leq h_i^{r,\text{in}}(t_i) \leq 1, \tag{8c}$$

$$0 \leq h_i^r(t_i) = \lambda_i^r(t_i)/\mu_i \leq 1. \tag{8d}$$

Constraint (8b) ensures that the *expected* number of contents does not exceed the cache size. (8c) and (8d) are the inherent constraints on occupancy probability $h_i^{r,\text{in}}(t_i) = \hat{F}_i(t_i)$ and hit probability $h_i^r(t_i) = \lambda_i^r(t_i)/\mu_i = F_i(t_i)$, respectively. Although the objective function is concave, (8) is not a convex optimization problem w.r.t. timer t_i, since the feasible set is not convex. See Appendix X-B for details. Hence, (8) is hard to solve in general.

In the following, we will show that (8) can be reformulated as a convex problem. From (5), we have $t_i = F_i^{-1}(\lambda_i^r/\mu_i)$, with $F_i^{-1}(\cdot)$ being the inverse function of $F_i(\cdot)$. Then by (6),

$$h_i^{r,\text{in}} = \hat{F}_i(F_i^{-1}(\lambda_i^r/\mu_i)) \triangleq g_i(\lambda_i^r/\mu_i). \tag{9}$$

From (3), we know there exists a one-to-one correspondence between $\hat{F}_i$ and F_i, hence $g_i(\cdot)$ exists. Therefore, (8) can be reformulated as follows

$$\textbf{HRB-CUM:} \quad \max_{\{\lambda_1,\cdots,\lambda_n\}} \quad \sum_{i=1}^{n} U_i(\lambda_i^r) \tag{10a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} g_i(\lambda_i^r/\mu_i) \leq B, \tag{10b}$$

$$0 \leq \lambda_i^r/\mu_i \leq 1. \tag{10c}$$

Again (10b) is on average cache occupancy. Note that we can obtain HPB-CUM from (10) by replacing λ_i^r by h_i^p in (10a), and λ_i^r/μ_i by h_i^p in (10b) and (10c), respectively.

Remark 1. *Let the buffer size $B(n)$ be a function of n and let ϵ be a constant greater than zero, $\epsilon > 0$. If $\epsilon^2 B(n) = \omega(1)$, then the probability that the number of cached contents exceeds $B(n)(1+\epsilon)$ decreases exponentially as a function of $\epsilon^2 B(n)$, [7]. Thus, we can let ϵ go to zero while allowing B to grow with n. The practical import is that the buffer can be sized as $B(1+\epsilon)$ while the optimizer works with B. Hence the fraction of buffer used, $\epsilon/(1+\epsilon)$, to protect against violations goes to zero as n gets large.*

A related problem has been formulated in [10], where the authors formulated the optimization problem as a function of $h_i^{r,\text{in}}$. However, such a formulation may not be suitable for designing decentralized algorithms since we need a closed form expression for $\hat{F}_i^{-1}$. More details on the advantages of our formulation over [10] in decentralized algorithm design are given in Section VI. Furthermore, [10] only considers linear utilities while we aim to characterize the impact of different utility functions on optimal TTL policies.

Now we consider the convexity of (10).

Lemma 1. *Let $F_i(t)$ and $\hat{F}_i(t)$ be the c.d.f. and age distribution for the request process of content i, given in (1) and (3), respectively. Denote the corresponding density function as $f_i(t)$. Let $\zeta_i(t)$ be the hazard rate function associated with $F_i(t)$, given as*

$$\zeta_i(t) = \frac{f_i(t)}{1 - F_i(t)}, \quad t \in [0, F_i^{-1}(1)]. \tag{11}$$

[1]From this section onwards, we will use superscript r and p to distinguish the corresponding hit rate, hit probability and occupancy probability under HRB-CUM and HPB-CUM, respectively.

Then we have

$$\frac{\partial g_i(\lambda_i^r/\mu_i)}{\partial \lambda_i^r} = \frac{1 - F_i(F_i^{-1}(\lambda_i^r/\mu_i))}{f_i(F_i^{-1}(\lambda_i^r/\mu_i))} = \frac{1}{\zeta_i(F_i^{-1}(\lambda_i^r/\mu_i))}. \tag{12}$$

The proof can be found in Appendix X-B.

From (12), it is clear that the behavior of the hazard rate function plays a prominent role in solving (10). In particular, if $\zeta_i(t)$ is a non-increasing hazard rate function (DHR), then by (12), $g'(\lambda_i^r/\mu_i)$ is non-decreasing in λ_i^r. Therefore, the feasible set in (10) is convex. Since the objective function is strictly concave and continuous, (10) is a convex optimization problem, and an optimal solution exists. In this paper, we mainly focus on the case that $\zeta_i(t)$ is DHR , and refer the interested reader to [10] for discussions of other cases. We will discuss several widely used distributions satisfying DHR in Section IV. In particular, we will also consider a uniform distribution, which has an increasing hazard rate, under which (10) is not a convex optimization problem, but we will show that an efficient approximate solution exists under linear or quadratic utilities.

In the following, we focus on the case that $\zeta_i(t)$ is DHR, i.e., (10) is a convex optimization problem. We write the Lagrangian function as

$$\mathcal{L}^r(\boldsymbol{\lambda}^r, \eta^r) = \sum_{i=1}^{n} U_i(\lambda_i^r) - \eta^r \left[\sum_{i=1}^{n} g_i(\lambda_i^r/\mu_i) - B\right], \tag{13}$$

where η^r is the Lagrangian multiplier and $\boldsymbol{\lambda}^r = (\lambda_1^r, \cdots, \lambda_n^r)$. We first consider complementary slackness conditions [33], i.e., $\eta^r[\sum_{i=1}^n g_i(\lambda_i^r/\mu_i) - B] = 0$. It is clear that $\eta^r \neq 0$, otherwise $\mathcal{L}^r(\boldsymbol{\lambda}^r, \eta^r)$ is maximized at $\lambda_i^r = \mu_i$, $\forall i$. Therefore, $\sum_{i=1}^n g_i(\lambda_i^r/\mu_i) = n \nleq B$, which does not satisfy the constraint.

To achieve the maximum of $\mathcal{L}^r(\boldsymbol{\lambda}^r, \eta^r)$, its derivative w.r.t. λ_i^r for $i = 1, \cdots, n$, should satisfy

$$\frac{\partial \mathcal{L}^r(\boldsymbol{\lambda}^r, \eta^r)}{\partial \lambda_i^r} = U_i'(\lambda_i^r) - \frac{\eta^r}{\mu_i} g_i'(\lambda_i^r/\mu_i) = 0, \tag{14}$$

i.e.,

$$\eta^r = \frac{\mu_i U_i'(\lambda_i^r)}{g_i'(\lambda_i^r/\mu_i)} \triangleq y_i(\lambda_i^r/\mu_i), \tag{15}$$

where $y_i(\cdot)$ is a continuous and differentiable function on $[0, 1]$. Hence we have

$$\lambda_i^r = \begin{cases} \mu_i y_i^{-1}(\eta^r), & 0 \leq y_i^{-1}(\eta^r) \leq 1, \\ \mu_i, & y_i^{-1}(\eta^r) > 1, \\ 0, & y_i^{-1}(\eta^r) < 0. \end{cases} \tag{16}$$

Again, by the cache capacity constraint, we can compute η^r through the following fixed-point equation

$$\sum_{i=1}^{n} g_i(\lambda_i^r/\mu_i) = \sum_{i=1}^{n} g_i(y_i^{-1}(\eta^r)) = B. \tag{17}$$

As discussed earlier, our optimization framework holds for TTL caches. Once we determine η^r from (17), the timer can be computed as

$$t_i = F_i^{-1}(y_i^{-1}(\eta^r)), \quad i = 1, \cdots, n, \tag{18}$$

then by (5), the hit probability and hit rate for reset TTL cache under HRB-CUM is

$$h_i^r = y_i^{-1}(\eta^r), \quad \lambda_i^r = \mu_i y_i^{-1}(\eta^r). \tag{19}$$

Remark 2. *Note that the above solution only requires the knowledge of irt distribution. It does not depend on any dependencies among inter-request times.*

IV. Specific Inter-request Time Distributions

In this section, we investigate irt distributions that are DHR such that (10) is a convex optimization problem. For ease of exposition, we relegate detailed explanations of different parameters and derivations to [28]. The properties of these distributions are presented in Table I.

First, for both exponential and generalized Pareto distributions, we have explicit forms for $g_i(\cdot)$. Thus the optimization problem in (10) can be solved both centrally and in a distributed manner with a distributed algorithm. However, we will see that the distributed dual algorithm for generalized Pareto distribution involves solving a fixed point equation, which has high computational complexity. This will be further discussed in Section VI.

Second, for hyperexponential distribution, we cannot obtain an explicit form for $F_i^{-1}(\cdot)$, and hence not for $g_i(\cdot)$ from (9). Therefore, it is difficult to obtain an exact solution of (10) through centralized solver besides a few special cases. However, we will see that the corresponding problems of (10) can be solved in a distributed fashion through solving fixed point equations without the explicit form of $g_i(\cdot)$. Again, this will be further discussed in Section VI.

An important class of processes that give rise to hyperexponential irt distributions are Markov modulated Poisson processes (MMPP). MMPP is a doubly stochastic Poisson process with rate varying according to a Markov process. MMPPs have been widely used to model request processes with bursty arrivals, which occur in various application domains such as web caching [31] and Internet traffic modeling [30]. We consider arrivals following a two state MMPP. W.l.o.g., denote the states as 1 and 2. The transition rate for content i from state 1 to 2 is r_{1i}, and r_{2i} vice versa. The arrivals for content i at states 1 and 2 are described by Poisson processes with rates θ_{1i} and θ_{2i}, respectively. Then the steady state distribution satisfies $\boldsymbol{\pi}_i = [\pi_{1i}, \pi_{2i}] = [r_{2i}/(r_{1i} + r_{2i}), r_{1i}/(r_{1i} + r_{2i})]$. Denote $\boldsymbol{p}_i = [p_{1i}, p_{2i}] = [\frac{\theta_{1i} r_{2i}}{\theta_{1i} r_{2i} + \theta_{2i} r_{1i}}, \frac{\theta_{2i} r_{1i}}{\theta_{1i} r_{2i} + \theta_{2i} r_{1i}}]$. We assume that the initial probability vector for this 2-MMPP is chosen according to $\boldsymbol{p}$, i.e., interval stationary [11]. Under this assumption, the inter-request times of this 2-MMPP is equivalent to a second order hyperexponential distribution with parameters satisfying [21]

$$u_{1i} = (\theta_{1i} + \theta_{2i} + r_{1i} + r_{2i} - \delta_i)/2,$$

Processes	Parameters	$\mathbf{F_i(t)}$	$\hat{\mathbf{F}}_\mathbf{i}(\mathbf{t})$	$\mathbf{g_i(x)}$	Optimal Solution
Process with exponential irt	μ_i: rate	$1-e^{-\mu_i t}$	$1-e^{-\mu_i t}$	x	centralized: convex solver decentralized: Dual
Process with generalized Pareto irt	k_i : shape, σ_i: scale $\theta_i(=0)$: location	$1-(1+\frac{k_i t}{\sigma_i})^{-\frac{1}{k_i}}$	$1-(1+\frac{k_i t}{\sigma_i})^{\frac{k_i-1}{k_i}}$	$1-(1-x)^{1-k_i}$	centralized: convex solver decentralized: Dual + fixed point
Process with hyper-exponential irt	l: order p_{ji} : phase probability θ_{ji} : phase rate	$1-\sum_{j=1}^{l} p_{ji}e^{-\theta_{ji}t}$	$\mu_i \sum_{j=1}^{l} \frac{p_{ji}}{\theta_{ji}}(1-e^{-\theta_{ji}t})$	No closed form	centralized: No exact solution decentralized: Dual + fixed point
2-MMPP Process	θ_{1i},θ_{2i} : arrival rate r_{1i},r_{2i} : Tran. rate $q_{1i},q_{2i},u_{1i},u_{2i}$: (20)	$1-\sum_{j=1}^{2} q_{ji}e^{-u_{ji}t}$	$\mu_i \sum_{j=1}^{2} \frac{q_{ji}}{u_{ji}}(1-e^{-u_{ji}t})$	No closed form	centralized: No exact solution decentralized: Dual + fixed point

TABLE I: Properties of specific irt distributions. The final column entitled "Optimal Solution" will be discussed in Section VI, where "centralized" is obtained by solving (10) and "decentralized" is obtained through designing decentralized algorithms.

$$
\begin{aligned}
u_{2i} &= (\theta_{1i}+\theta_{2i}+r_{1i}+r_{2i}+\delta_i)/2,\\
q_{1i} &= \frac{\theta_{2i}^2 r_{1i}+\theta_{1i}^2 r_{2i}}{(\theta_{1i}r_{2i}+\theta_{2i}r_{1i})(u_{1i}-u_{2i})} - \frac{u_{2i}}{u_{1i}-u_{2i}},\\
q_{2i} &= 1-q_{1i},\\
\delta_i &= \sqrt{(\theta_{1i}-\theta_{2i}+r_{1i}-r_{2i})^2+4r_{1i}r_{2i}}.
\end{aligned} \tag{20}
$$

Again, it is difficult to obtain exact solution of (10) through centralized solver for a two state MMPP.

Remark 3. *Note that obtaining optimal solution for inter-request times with hyperexponential distribution has significant advantages since many heavy-tailed distributions can be well approximated by a hyperexponential distribution [9]. Similarly for an m-state MMPP, $F_i(\cdot)$ is a mixture of m exponential distributions. In particular, we will discuss the algorithm for obtaining optimal solution for a two-state MMPP in Section VII. Furthermore, we also consider Weibull distribution. Due to space limits, we relegate its property to [28].*

V. Performance Comparison

Different utility functions define different fairness properties. In this section, we analytically compare the performance of HRB-CUM and HPB-CUM under different utility functions and request arrival processes considered in Section IV. We omit proofs in this section and relegate them to Appendix X-C.

A. Identical Distributions

Assume that all contents have the same request arrival process, i.e., $F_i(\cdot)=F(\cdot)$ for all i, then we have $\hat{F}_i(\cdot)=\hat{F}(\cdot)$, $g_i(\cdot)=g(\cdot)$ and $\mu_i=\mu$ for all i.

Theorem 1. *Under identical stationary request processes, HRB-CUM (8) and HPB-CUM* (33) *in Appendix X-A are equivalent.*

Further assume that all contents have the same utility function, i.e., $U_i(\cdot)=U(\cdot)$, for all i. From (15), we know $y_i^{-1}=y^{-1}$ for all i. Hence $\lambda_i^r=\lambda^r$ for all i. Therefore, from (17),

$$
\sum_{i=1}^{n} g_i(\lambda_i^r/\mu_i) = n g(\lambda^r/\mu) = B, \quad \text{i.e.,} \quad \lambda^r = \mu g^{-1}(B/n). \tag{21}
$$

B. β-fair Utility Functions

Consider the case that $\beta=1$ in (7), i.e., $U_i(x)=w_i\log x$, then $U_i'(x)=w_i/x$.

Theorem 2. *HRB-CUM (8) and HPB-CUM* (33) *in Appendix X-A are identical under β-utility function with $\beta=1$.*

In the remainder of this section, we consider β-fair utility function with $\beta>0$ and $\beta\neq 1$. We compare the optimal hit probabilities h_i^r, h_i^p and hit rates λ_i^r, λ_i^p, under HRB-CUM (8) and HPB-CUM (33) in Appendix X-A for different weights w_i. Without loss of generality (w.l.o.g.), we assume arrival rates satisfy $\mu_1\geq\cdots\geq\mu_n$, such that content popularities satisfy $p_1\geq\cdots\geq p_n$, where $p_i=\mu_i/\mu$ and $\mu=\sum_i\mu_i$.

1) Poisson Request Processes: With the Lagrangian method, we easily obtain the optimal hit rate λ_i^r and hit probability h_i^r under HRB-CUM for $\beta>0$ and $\beta\neq 1$,

$$
\lambda_i^r = \frac{w_i^{1/\beta}\mu_i^{1/\beta}}{\sum_j w_j^{1/\beta}\mu_j^{1/\beta-1}}B, \quad h_i^r = \frac{w_i^{1/\beta}\mu_i^{1/\beta-1}}{\sum_j w_j^{1/\beta}\mu_j^{1/\beta-1}}B. \tag{22}
$$

From [7], the corresponding optimal hit rate and hit probability under HPB-CUM are $\lambda_i^p=\frac{w_i^{1/\beta}\mu_i}{\sum_j w_j^{1/\beta}}B$ and $h_i^p=\frac{w_i^{1/\beta}}{\sum_j w_j^{1/\beta}}B$, respectively. Consider the Zipf popularity distribution with parameter $\alpha=0.8$, $n=10^3$ and $B=100$ in our numerical studies.

Monotone non-increasing weights: We consider monotone non-increasing weights, i.e., $w_1\geq\cdots\geq w_n$, given $\mu_1\geq\cdots\geq\mu_n$.

Theorem 3. *When $\{w_i, i=1,\cdots,n\}$ are monotone decreasing, (i) for $\beta<1$, $\exists\tilde{j}\in(1,n)$ s.t. $\lambda_i^r>\lambda_i^p$, $\forall i<\tilde{j}$; and (ii) for $\beta>1$, $\exists\tilde{l}\in(1,n)$ s.t. $\lambda_i^r>\lambda_i^p$, $\forall i>\tilde{l}$. In particular, if $\tilde{j},\tilde{l}\in\mathbb{Z}^+$, then $\lambda_{\tilde{j}}^r=\lambda_{\tilde{j}}^p$, and $\lambda_{\tilde{l}}^r=\lambda_{\tilde{l}}^p$.*

Theorem 3 states that compared to HPB-CUM, HRB-CUM favors more popular contents for $\beta<1$, and less popular contents for $\beta>1$.

The following corollary applies to the Zipf popularity distribution.

Corollary 1. *If the popularity distribution is Zipfian: (a) When $\beta<1$, $\lambda_i^r>\lambda_i^p$ for $i=1,\cdots,i_0$, and $\lambda_i^r<\lambda_i^p$ for $i=i_0+1,\cdots,n$; (b) When $\beta>1$, $\lambda_i^r<\lambda_i^p$ for*

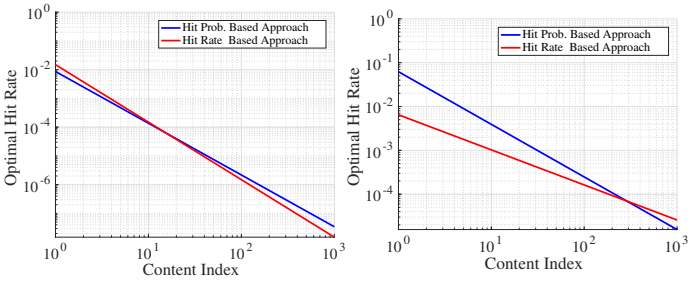

Fig. 1: HRB-CUM vs. HPB-CUM under exponential distribution: *(Left)* $\beta = 0.8$ and *(Right)* $\beta = 2$.

$i = 1, \cdots, i_0$, *and* $\lambda_i^r > \lambda_i^p$ *for* $i = i_0 + 1, \cdots, n$, *where* $i_0 = \left\lfloor \left(\frac{\sum_j w_j^{\frac{1}{\beta}} j^{\alpha(1-\frac{1}{\beta})}}{\sum_j w_j^{\frac{1}{\beta}}} \right)^{\frac{1}{\alpha(1-\frac{1}{\beta})}} \right\rfloor$.

Figures 1 *(Left)* and *(Right)* illustrate the case that $w_i = \mu_i$, $\beta = 0.8$ and $\beta = 2$, respectively.

We make a similar comparison of the hit probabilities under HRB-CUM and HPB-CUM.

Theorem 4. *When* $w_1 \geq \cdots \geq w_n$, *(i) for* $\beta < 1$, $\exists j \in (1, n)$ *s.t.* $h_i^r > h_i^p$, $\forall i < j$, *and* $h_i^r < h_i^p$, $\forall i > j$; *and (ii) for* $\beta > 1$, $\exists l \in (1, n)$ *s.t.* $h_i^r < h_i^p$, $\forall i < l$, *and* $h_i^r > h_i^p$, $\forall i > l$. *In particular, if* $j, l \in \mathbb{Z}^+$, *then* $h_j^r = h_j^p$, *and* $h_l^r = h_l^p$.

The following corollary applies to the Zipf popularity distribution.

Corollary 2. *Assume the Zipf popularity distribution: (a) When* $\beta < 1$, $h_i^r > h_i^p$, *for* $i = 1, \cdots, i_0$, *and* $h_i^r < h_i^p$, *for* $i = i_0 + 1, \cdots, n$; *(b) When* $\beta > 1$, $h_i^r < h_i^p$, *for* $i = 1, \cdots, i_0$, *and* $h_i^r > h_i^p$, *for* $i = i_0 + 1, \cdots, n$, *where* $i_0 = \left\lfloor \left(\frac{\sum_j w_j^{\frac{1}{\beta}} j^{\alpha(1-\frac{1}{\beta})}}{\sum_j w_j^{\frac{1}{\beta}}} \right)^{\frac{1}{\alpha(1-\frac{1}{\beta})}} \right\rfloor$.

We numerically verify our results, and observe that they exhibit similar trends as in Figures 1 *(Left)* and *(Right)*, hence we omit them here due to space constraints.

We are unable to achieve explicit expressions for h_i^r, h_i^p, λ_i^r and λ_i^p for HRB-CUM and HPB-CUM when inter-request times follow either a generalized Pareto or hyperexponential distribution, or 2-MMPP process. However, from Section IV, we know (10) and (33) in Appendix X-A are convex optimization problems. We numerically compare the performance of HRB-CUM (8) and HPB-CUM (33) in Appendix X-A under a Zipf-like distribution with parameter $\alpha = 0.8$. Similar results hold for these distributions, and we omit the results due to space limitation.

VI. Decentralized Algorithms

In Section III, we formulated the optimization problem with a fixed cache size, however, system parameters (e.g. request processes) can change over time, and as discussed in Section IV, the optimization problem under some inter-request distributions cannot be solved centrally. Moreover, it is infeasible to solve the optimization problem offline and then implement the optimal strategy. Hence decentralized algorithms are needed to implement the optimal strategy to adapt to these changes in the presence of limited information. In the following, we develop such algorithms for HRB-CUM and compare its performance to that of HPB-CUM under stationary request processes discussed in Section IV. We only present explicit algorithms for HRB-CUM, similar algorithms for HPB-CUM are available in Appendix X-D. We drop the superscript r in this section for brevity.

A. Dual Algorithm

For a request arrival process with a DHR inter-request distribution, (10) becomes a convex optimization problem as discussed in Section IV, and hence solving the dual problem produces the optimal solution. Since $0 < t_i < \infty$, then $0 < \lambda_i/\mu_i < 1$ and $0 < g_i(\lambda_i/\mu_i) < 1$. Therefore, the Lagrange dual function is

$$D(\eta) = \max_{\lambda_i} \left\{ \sum_{i=1}^{n} U_i(\lambda_i) - \eta \left[\sum_{i=1}^{n} g_i\left(\lambda_i/\mu_i\right) - B \right] \right\}, \tag{23}$$

and the dual problem is

$$\min_{\eta \geq 0} \quad D(\eta). \tag{24}$$

Following the standard *gradient descent algorithm* by taking the derivative of $D(\eta)$ w.r.t. η, the dual variable η should be updated as

$$\eta^{(k+1)} \leftarrow \max \left\{ 0, \eta^{(k)} + \gamma \left[\sum_{i=1}^{n} g_i(\lambda_i^{(k)}/\mu_i) - B \right] \right\}, \tag{25}$$

where k is the iteration number, $\gamma > 0$ is the step size at each iteration and $\eta \geq 0$ due to KKT conditions.

Based on the results in Section III, in order to achieve optimality, we must have

$$\eta^{(k)} = \frac{\mu_i U_i'(\lambda_i^{(k)})}{g_i'(\lambda_i^{(k)}/\mu_i)} \triangleq y_i(\lambda_i^{(k)}/\mu_i), \text{ i.e., } \lambda_i^{(k)} = \mu_i y_i^{-1}(\eta^{(k)}). \tag{26}$$

Since $g_i(\lambda_i^{(k)}/\mu_i)$ indicates the probability that content i is in the cache, $\sum_{i=1}^{n} g_i(\lambda_i^{(k)}/\mu_i)$ represents the number of contents currently in the cache, denoted as B_{curr}. Therefore, the dual algorithm for reset TTL caches is

$$t_i^{(k)} = F_i^{-1}(y_i^{-1}(\eta^{(k)})), \tag{27a}$$

$$\eta^{(k+1)} \leftarrow \max \left\{ 0, \eta^{(k)} + \gamma(B_{\text{curr}} - B) \right\}, \tag{27b}$$

where the iteration index k is incremented upon each request.

Remark 4. *From (26) and (27), it is clear that if the explicit form of* $g_i(\cdot)$ *or* $g_i'(\cdot)$ *is available, then the dual algorithm can be directly implemented. This is the case for Poisson and generalized Pareto inter-request distributions, see Section IV and the following for details. However, neither is available for the hyberexponential distribution and the* 2*-MMPP. In the following, we will show that the dual algorithm can still be implemented without this information.*

Poisson Process: We have $g_i'(\lambda_i^{(k)}/\mu_i) = 1$, and $\lambda_i^{(k)} = U_i'^{-1}(\eta^{(k)}/\mu_i)$.

Generalized Pareto Distribution: When inter-request times are described by a generalized Pareto distribution and utilities are β-fair, $\lambda_i^{(k)}$ can be obtained through

$$\mu_i^{1-\beta} w_i (1-(\lambda_i^{(k)}/\mu_i))^{k_i}/[\eta^{(k)}(1-k_i)] - (\lambda_i^{(k)}/\mu_i)^{\beta} = 0. \tag{28}$$

We can show that there exists a finite value solution in $[0, \mu_i]$ for any $\eta^{(k)} > 0$; details given in Appendix X-D1.

Hyperexponential Distribution: Under a hyperexponential distribution, we have $g_i'(x) = \mu_i(1-x)/f_i(F_i^{-1}(x))$. Since we do not have a closed form expression for $F_i^{-1}(x)$, no explicit form exists for $g_i'(x)$. Given (26) and a β-fair utility, timer $t_i^{(k)}$ at iteration k can be solved from the following fixed point equation

$$(F_i(t_i^{(k)}))^{\beta+1} - (F_i(t_i^{(k)}))^{\beta} + f_i(t_i^{(k)})/[\eta^{(k)}\mu_i^{\beta-1}] = 0, \tag{29}$$

where $F_i(t) = 1 - \sum_{j=1}^{l} p_{ji} e^{-\theta_{ji} t}$ and $f_i(t) = \sum_{j=1}^{l} p_{ji}\theta_{ji} e^{-\theta_{ji} t}$.

2*-MMPP:* From Section IV, 2-MMPP is equivalent to a second order hyperexponential distribution. Hence timer $t_i^{(k)}$ can be updated from (29) with $F_i(t) = 1 - \sum_{j=1}^{2} q_{ji} e^{-u_{ji} t}$ and $f_i(t) = \sum_{j=1}^{2} q_{ji} u_{ji} e^{-u_{ji} t}$.

Remark 5. *We can similarly design primal and primal-dual algorithms by introducing a cost function $C(\cdot)$ to the sum of utilities, which is a convex and non-decreasing penalty function denoting the cost for extra cache storage. For ease of exposition, we relegate their description to [28]. In the remainder of the paper, we refer to these distributed algorithms as Dual, Primal and Primal-Dual, respectively. Furthermore, it can be easily shown that all the above distributed algorithms converge to the optimal solutions, respectively, using Lyapunov techniques. Again, the corresponding algorithms for Weibull distribution are available in [28].*

B. Performance Evaluation

In this Section, we evaluate the performance of the decentralized algorithms for both HRB-CUM and HPB-CUM when inter-request times are described by stationary request processes with an exponential irt distribution when utility functions are β-fair. Due to space restrictions, we limit our study to minimum potential delay fairness, i.e., $\beta = 2$.

1) Experiment Setup: In our studies, we consider a Zipf popularity distribution with $\alpha = 0.8$, $n = 1000$ and $B = 100$. We consider irt distributions described in Section IV with an aggregate request rate $\mu = 1$ such that $\mu_i = p_i$ from (4). In particular, for exponential distribution, the rate parameter is set to $\mu_i = p_i$. We relegate the discussions of generalized Pareto, hyperexponential and 2-MMPP to Section VII

2) Exactness: We first consider the dual algorithm described in Section VI-A. Note that the dual algorithm for generalized Pareto involves solving singular equation (28). We solve it efficiently with Matlab routine *fsolve* using a step size[2] $\gamma = 10^{-7}$. The performance of dual under exponential is shown in Figure 2 *(Left)*, where "Centralized" means solutions from solving (10).

From Figure 2 *(Left-Top)*, we observe that the decentralized algorithms yield the exact hit rates under both HRB-CUM and HPB-CUM. Similarly results hold for the hit probabilities, which is omitted here due to space limits. Figure 2 *(Left-Bottom)* shows the probability density for the number of contents in the cache across these distributions. As expected the density is highly concentrated around the cache size B. Similar results hold for generalized Pareto distribution and we omit the results due to space limits.

We also use primal and primal-dual distributed algorithms to implement minimum potential delay fairness. In particular, as discussed in Section VI, primal is associated with a penalty function $C(\cdot)$. Choosing an appropriate penalty function plays an important role in the performance of primal, since we need to evaluate the gradient at each iteration through $C'(\cdot)$. Here, we use $C(x) = \max\{0, x - B\log(B+x)\}$ [33]. Another reasonable choice can be $C(x) = \max\{0, x^m\}$, $m \geq 1$. We observe that both primal and primal-dual yield exact hit probabilities and hit rates under HRB-CUM and HPB-CUM for minimum potential delay fairness. We omit the plots due to space constraints.

3) Convergence Rate and Robustness: Although the decentralized algorithms converge to the optimal solution as shown in Section VI-B2, the rate of convergence is also important from a service provider's perspective. Due to space limits, we only focus on the dual here. From (27), it is clear that the step size[3] γ^p (or γ^r) plays a significant role in the convergence rate. We choose different values of γ^p and γ^r and compare the performance of HRB-CUM and HPB-CUM under Poisson request processes, shown in Figure 2 *(Middle)* and *(Right)*. On one hand, we find that when a larger value of $\gamma^p = \gamma^r = 10^{-3}$ is chosen, the dual for HRB-CUM easily converges after a few iterations (around 5×10^5 iterations), i.e., the simulated hit rates exactly match numerically computed values, while those of HPB-CUM do not converge. On the other hand, when a smaller value $\gamma^p = \gamma^r = 10^{-5}$ is chosen, both converge in the same number of iterations. We also used $\gamma^p = \gamma^r = 10^{-1}, 10^{-7}$, which exhibit similar behaviors to 10^{-3} and 10^{-5}, respectively, and are omitted due to space constraints.

We also explored the expected number of contents in the cache, shown in Figure 2 *(Middle)* and *(Right)*. It is obvious that under HRB-CUM, the probability of violating the target cache size B is quite small, while it is larger for HPB-CUM especially for $\gamma^p = \gamma^r = 10^{-3}$, and even for $\gamma^p = \gamma^r = 10^{-5}$,

[2]Note that the step size has an impact on the convergence and its rate, more details are discussed in Section VI-B3.

[3]Here we use superscript p and r to distinguish the step size of corresponding dual algorithms under HRB-CUM and HPB-CUM, respectively.

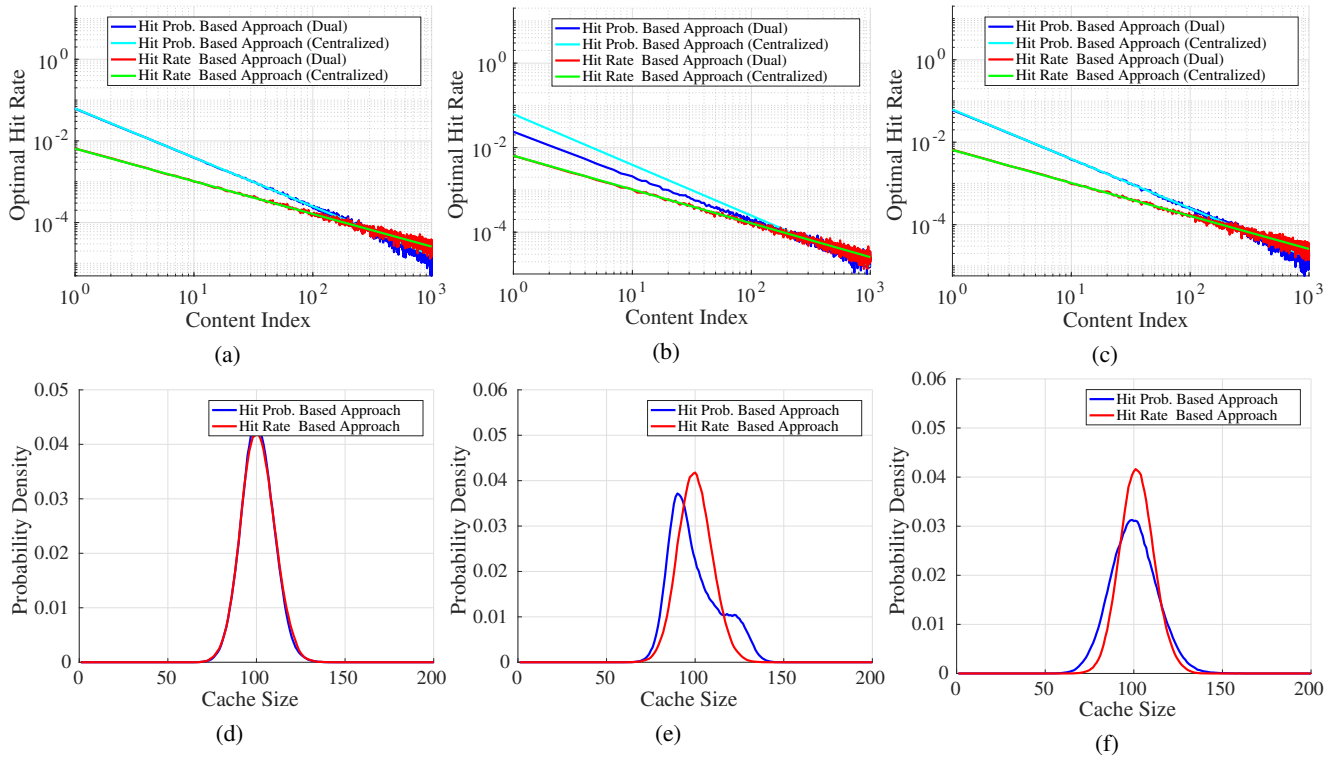

Fig. 2: Dual for HRB-CUM and HPB-CUM with under exponential distribution with minimum potential delay fairness; Hit rate (Fig. *a-c*) and cache size distribution (Fig. *d-f*) comparisons for Dual with $\gamma = 10^{-7}$ *(left)*, $\gamma = 10^{-3}$ *(Middle)* and $\gamma = 10^{-5}$ *(Right)* under exponential inter-request process.

HRB-CUM is more concentrated on the target size B. These results indicate that the dual algorithm associated with HRB-CUM is more robust to changes in the step size and converges much faster under exponential inter-requests.

4) Comparison of Decentralized Algorithms: From the above analysis, we know that at each iteration, the dual algorithm needs to solve a non-linear equation to obtain a timer value, which might be computationally intensive compared to primal and primal-dual. However, for primal, some choices of penalty function $C(\cdot)$ and arrival process $g_i(\cdot)$ may result in large gradients and abrupt function change [32]. Similarly for primal-dual, two scaling parameters δ_i and γ need to be carefully chosen, otherwise the algorithm might diverge. These demonstrate the pro-and-cons of these distributed algorithms, and one algorithm may be favorable than others in specific situations.

VII. Poisson Online Approximation

From Section VI, it is clear that the implementation of Dual under generalized Pareto, hyperexponential distributions and 2-MMPP involves solving non-linear fixed point equations, which are computationally intensive. However, the Dual for the case of requests governed by Poisson processes is simple. Furthermore, knowledge of the irt distribution are also required. However, this is not always available to the service provider.

In this section, we apply the Dual designed for the case requests are described by Poisson processes to a workload where requests are described by a *n*on-Poisson stationary request processes. Such an algorithm does not require solving non-linear equations and hence is computationally efficient. Moreover, we also use estimation techniques introduced in [7] to approximate request rates which makes these distributed algorithms work in an online fashion.

A. Online Algorithm

We consider the problem of estimating the arrival rate μ_i for content i adopting techniques used in [7] described as follows. Denote the remaining TTL time for content i as τ_i. This can be computed given t_i and a time-stamp for the last request time for content i. Recall that X_{ik} is a random variable corresponding to the inter-request times for requests for content i. Let $\bar{X}_{ik}$ be the mean. Then we approximate the mean inter-request time as $\hat{\bar{X}}_{ik} = t_i - \tau_i$. Clearly $\hat{\bar{X}}_{ik}$ is an unbiased estimator of $\bar{X}_{ik}$, and hence an unbiased estimator of $1/\mu_i$. In this section, we use this estimator to implement the distributed algoritms, which now becomes an online algorithm.

Given this estimator and Dual (27), we propose the following *Poisson approximate online algorithm*

$$t_i^{(k)} = -\frac{1}{\hat{\mu}_{iP}} \log\left(1 - \frac{1}{\hat{\mu}_{iP}} U_i'^{-1}\left(\frac{\eta^{(k+1)}}{\hat{\mu}_{iP}}\right)\right), \tag{30a}$$

$$\eta^{(k+1)} \leftarrow \max\{0, \eta^{(k)} + \gamma(B_{\text{curr}} - B)\}. \tag{30b}$$

There are two differences between our proposed algorithm (30) and Dual (27). First, the explicit form of (27a) is different for different inter-request distributions as discussed in Section VI-A, while we always adopt the explicit form of Poisson process in (30a). Second, μ_i in (26) is the exact value

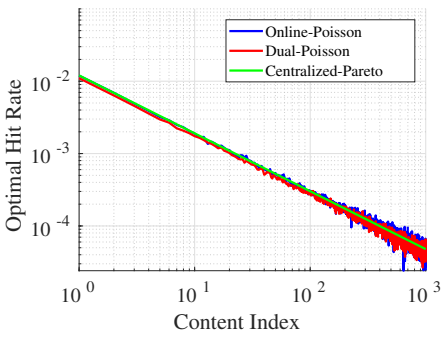

Fig. 3: Poisson online approximation to Generalized Pareto inter-requests.

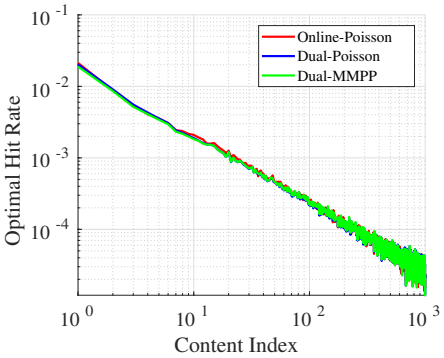

Fig. 4: Poisson online approximation to 2-MMPP inter-requests: $x = 10^{-3}$.

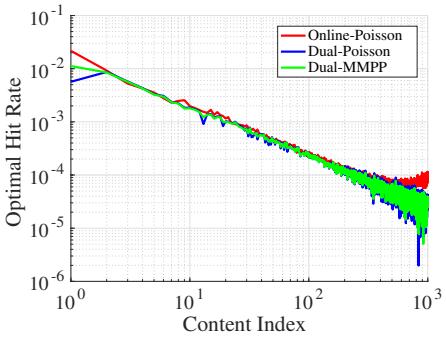

Fig. 5: Poisson online approximation to 2-MMPP inter-requests: $x = 10^{-7}$.

of the mean arrival rate of the corresponding inter-request distribution, while we estimate its value as discussed above and denote it as $\hat{\mu}_{iP}$. However, the value of B_{curr} denotes the number of contents currently in the cache under the real inter-request distribution under both (30) and (27). In the following, we consider the performance of (30) under different inter-request distributions.

B. Generalized Pareto Distribution

In this section, we apply the online algorithm (30) to a workload where requests are described by stationary request process under generalized Pareto distribution with shape parameter $k_i = 0.48$. The performance is shown in Figure 3 . It is clear that the approximation is accurate. Furthermore, it has been theoretically characterized in [34] that for any given generalized Pareto model with finite variance, an optimal exponential approximation exists which minimizes the K-L divergence between these two distributions. The optimal exponential approximation has the same mean as that of the generalized Pareto distribution, i.e. $\mu_i = (1 - k_i)/\sigma_i$. The estimator we use in our online algorithm (30), i.e., $1/\hat{\mu}_{iP}$, is an unbiased estimator of mean inter-request rate of the generalized Pareto arrival process, thus explaining the better performance of our Poisson approximation in accordance with the theoretical results provided in [34]. Moreover, we notice that when k_i becomes smaller, the accuracy has been improved. However, this approximation has poor performance when $k_i > 0.5$ since the generalized Pareto distribution has infinite variance for $k_i > 0.5$.

C. 2-MMPP

The optimal hit rates under 2-MMPP can be obtained through solving Dual for a second order hyperexponential distribution with parameters q, u_1 and u_2 defined in (20). However, from Section VI, Dual requires solving a non-linear equation (29). Instead, we consider Poisson approximation (30) under 2-MMPP. W.l.o.g., we assume the phase rates θ_{1i} and θ_{2i} for $i = 1, \cdots, n$ to be Zipf distributed with parameters 0.4 and 0.8, respectively.

1) Limiting Behavior: We first evaluate the performance of Poisson online approximation algorithm (30) for different transition rates r_{1i} and r_{2i}. W.l.o.g., we represent $r_{1i} = a_{1i}x_i$ and $r_{2i} = a_{2i}x_i$, where a_{1i}, a_{2i} are constants, and $0 \leq x_i \leq \infty$.

Theorem 5. *We represent $r_{1i} = a_{1i}x_i$ and $r_{2i} = a_{2i}x_i$, where a_{1i}, a_{2i} are constants, and $0 \leq x_i \leq \infty$. Then we have the following results*
(1) When $r_{1i}, r_{2i} \to \infty$, i.e., $x_i \to \infty$, by applying L'Hospital's rule to (20), we have

$$u_{1i} = \frac{\theta_{1i}a_{2i} + \theta_{2i}a_{1i}}{a_{1i} + a_{2i}}, \qquad u_{2i} = \infty,$$
$$q_{1i} = 1, \qquad q_{2i} = 0, \tag{31}$$

i.e., 2-MMPP is equivalent to a Poisson process with rate u_{1i}, i.e., our approximation is exact.
(2) When $r_{1i}, r_{2i} \to 0$, i.e., $x_i \to 0$, by applying L'Hospital's rule to (20), we have

$$u_{1i} = \theta_{2i}, \qquad u_{2i} = \theta_{1i},$$
$$q_{1i} = \frac{\theta_{2i}a_{1i}}{\theta_{1i}a_{2i} + \theta_{2i}a_{1i}}, \qquad q_{2i} = \frac{\theta_{1i}a_{2i}}{\theta_{1i}a_{2i} + \theta_{2i}a_{1i}}, \tag{32}$$

i.e., 2-MMPP can be treated as a weighted sum of two Poisson processes with rates θ_{1i}, θ_{2i} and weights q_{1i}, q_{2i}, respectively.
(3) When $0 < r_{1i}, r_{2i} < \infty$, i.e., $0 < x_i < \infty$: As discussed in Section IV, 2-MMPP is equivalent to a second order hyperexponential distribution with parameters given in (20).

The proof is relegated to Appendix X-F.

2) Numerical Validation: We numerically verify the results in Theorem 5 by taking different values of transition rates. The performance comparison between (31) and (32) are shown in Figures 4 and 5, respectively,where "Dual-MMPP" is obtained from Dual (27) in Section VI, "Dual-Poisson" is obtained from (30) with the exact mean $\mu_i = (\theta_{1i}r_{2i} + \theta_{2i}r_{1i})/(r_{1i} + r_{2i})$ is known and "Online-Poission" is obtained from (30) with estimated arrival rates as discussed in Section VII-A. We can see that with large transition rates, the poisson approximation performs better as compared to small transition rates. This is due to the fact that our approximation becomes exact when transition rates go to infinite. However, our approximation yields similar optimal aggregate hit rate as compared to "Dual-MMPP" even for small transition rates as shown in Table II. We also numerically verify the third case in Theorem 5 by taking $r_{1i} = 5 \times 10^{-5}$ and $r_{2i} = 2 \times 10^{-5}$. Again, we can see

x	n	B	Dual-MMPP	Dual-Poisson	Online-Poisson
10^{-3}	1000	100	0.1591	0.1612	0.1655
10^{-7}	1000	100	0.1474	0.1427	0.1540

TABLE II: Optimal aggregate hit rates for large ($x = 10^{-3}$) and small ($x = 10^{-7}$) state transition rates.

that the optimal hit rates obtained through (30) match those obtained from Dual under 2-MMPP. We omit the plot due to space limits.

Remark 6. *We also considered the case when irts follow hyperexponential and weibull distributions. Equation (10) can be solved with Dual for both distributions. We compare results using (10) with those obtained using (30) and we find that the optimal hit rates obtained through (30) match those obtained solving (10). For ease of exposition, these results are relegated to Appendix X-E.*

VIII. Conclusion

In this paper, we associated each content with a utility that is a function of the corresponding content hit rate or hit probability, and formulated a cache utility maximization problem under stationary requests. We showed that this optimization problem is convex when the request process has a DHR. We presented explicitly optimal solutions for HRB-CUM and HPB-CUM, and made a comparison between them both theoretically and numerically. We also developed decentralized algorithms to implement the optimal policies. We found that HRB-CUM is more robust and stable than HPB-CUM w.r.t. convergence rate. Finally, we proposed Poisson approximate online algorithms to different inter-request distributions, which is accurate and lightweight. Going further, we aim at extending our results to consider *Non-reset TTL Cache* where the timer is set only on a cache miss. Non-reset TTL Caches might have different implications on the design and performance analysis of distributed and online algorithms. Establishing these results will be our future goal.

IX. Acknowledgment

This research was sponsored by the U.S. ARL and the U.K. MoD under Agreement Number W911NF-16-3-0001 and by the NSF under Grant CNS-1617437. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation, U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. The authors also thank Dr. Bo Jiang for useful discussions on MMPP.

X. Appendix

A. HPB-CUM

Following a similar argument in Section III, we can formulate the following hit probability based optimization problem

$$\textbf{HPB-CUM:} \quad \max_{0\le h_i^p\le 1} \sum_{i=1}^{n} U_i(h_i^p), \quad \text{s.t.} \sum_{i=1}^{n} g_i(h_i^p) \le B, \tag{33}$$

The Lagrangian function can be written as

$$\mathcal{L}^p(\boldsymbol{h}^p, \eta^p) = \sum_{i=1}^{n} U_i(h_i^p) - \eta^p \left[\sum_{i=1}^{n} g_i(h_i^p) - B\right], \tag{34}$$

where η^p is the Lagrangian multiplier and $\boldsymbol{h}^p = (h_1^p, \cdots, h_n^p)$. Similarly, the derivative of $\mathcal{L}^p(\boldsymbol{h}^p, \eta^p)$ w.r.t. h_i^p for $i = 1, \cdots, n$, should satisfy the following condition so as to achieve its maximum

$$\eta^p = U_i'(h_i^p)/g_i'(h_i^p) \triangleq v_i(h_i^p), \tag{35}$$

where $v_i(\cdot)$ is a continuous and differentiable function on $[0, 1]$, i.e., there exists a one-to-one mapping between η^p and h_i^p if $0 \le v_i^{-1}(\eta^p) \le 1$. Again, by the cache capacity constraint, we can compute η^p through the following fixed-point equation

$$\sum_{i=1}^{n} g_i(h_i^p) = \sum_{i=1}^{n} g_i(v_i^{-1}(\eta^p)) = B. \tag{36}$$

Finally, given η^p, the timer, hit probability, and hit rate are

$$t_i = F_i^{-1}(v_i^{-1}(\eta^p)), \quad h_i^p = v_i^{-1}(\eta^p), \quad \lambda_i^p = \mu_i v_i^{-1}(\eta^p), \quad i = 1, \cdots, n. \tag{37}$$

B. Proofs in Section III

1) Convexity of HRB-CUM (8) and HPB-CUM (33): In this section, we show that HRB-CUM (8) and HPB-CUM (33) in terms of timers are non-convex.

Theorem. *HRB-CUM (8) and HPB-CUM (33) in terms of timers are non-convex.*

Proof. Recall that

$$\hat{F}_i(t_i) = \mu_i \int_0^{t_i} (1 - F(x))dx.$$

Take the derivative w.r.t. t_i, we have

$$\frac{\partial \hat{F}_i(t_i)}{\partial t_i} = \mu_i(1 - F(t_i)), \quad \text{and} \quad \frac{\partial^2 \hat{F}_i(t_i)}{\partial t_i^2} = -\mu_i f_i(t_i). \tag{38}$$

Since $f_i(\cdot)$ is the p.d.f. for the inter-request arrival time with $\mu_i \ge 0$, we have $f_i(t_i) \ge 0$. Thus $\partial^2 \hat{F}_i(t_i)/\partial t_i^2 \le 0$. Therefore, $\hat{F}_i(t_i)$ is concave in t_i and then (8) is a non-convex optimization problem. Similarly, we can show that (33) is non-convex. □

2) Proof for Lemma 1: Given (1) and (3), we have

$$\frac{\partial \hat{F}_i(t_i)}{\partial t_i} = \mu_i(1 - F_i(t_i)). \tag{39}$$

Then

$$\begin{aligned}\frac{\partial g_i(h_i^p)}{\partial h_i^p} &= \frac{\partial \hat{F}_i(F_i^{-1}(h_i^p))}{\partial h_i^p} \\ &\overset{(a)}{=} \mu_i(1 - F_i(F_i^{-1}(h_i^p))) \cdot \frac{\partial F_i^{-1}(h_i^p)}{\partial h_i^p} \\ &\overset{(b)}{=} \frac{\mu_i(1 - F_i(F_i^{-1}(h_i^p)))}{f_i(F_i^{-1}(h_i^p))} \\ &= \frac{\mu_i}{\zeta_i(F_i^{-1}(h_i^p))},\end{aligned} \tag{40}$$

where (a) and (b) hold true based on the chain-rule and the inverse function theorem over continuously differentiable function F_i, respectively.

C. Proofs in Section V

In this section, we compare the performance of HRB-CUM and HPB-CUM under different utility functions and inter-request processes.

1) Identical Distributions: Here, we consider the performance comparison of HRB-CUM and HPB-CUM under identical inter-request process.

Proof of Theorem 1 Under identical inter-request process, we have $F_i(\cdot) = F(\cdot)\ \forall i$. Hence $\hat{F}_i(\cdot) = \hat{F}(\cdot)$, i.e., $g_i(\cdot) = g(\cdot)\ \forall i$. Also $\mu_i = \mu\ \forall\ i$. In HRB-CUM (8), we aim to maximize the objective $\sum_{i=1}^n U_i(\lambda_i^r)$. We can scale the objective as $\sum_{i=1}^n U_i(\lambda_i^r/\mu)$, while the solution of problem (8) remains the same. By substituting $\lambda_i^r/\mu = h_i^p$, (8) and (33), i.e. HRB-CUM and HPB-CUM are identical.

2) β-fair Utility Functions: Here, we consider β-fair utilities. First, we consider log utilities, i.e., $\beta = 1$.

Proof for Theorem 2 Consider $U_i(x) = w_i \log x$, i.e., $U_i'(x) = w_i/x$. Under HRB-CUM, from (15), it is clear that

$$y_i(\lambda_i^r/\mu_i) = \frac{\mu_i w_i}{\lambda_i^r g_i'(\lambda_i^r/\mu_i)} = \frac{U_i'(\lambda_i^r/\mu_i)}{g_i'(\lambda_i^r/\mu_i)} = v_i(\lambda_i^r/\mu_i). \tag{41}$$

Again by substituting $\lambda_i^r/\mu_i = h_i^p$, HRB-CUM and HPB-CUM are identical.

Exponential Distribution: We compare HPB-CUM and HRB-CUM under exponential inter-request process.

Uniform weights: First we consider uniform weights, i.e., $w_i \equiv w$ for $i = 1, \cdots, n$. Then we have

$$\begin{aligned}h_i^p &= \frac{B}{n}, \\ h_i^r &= \frac{\mu_i^{\frac{1}{\beta}-1}}{\sum_{j=1}^n \mu_j^{\frac{1}{\beta}-1}} B.\end{aligned} \tag{42}$$

It is easy to check that h_i^r is decreasing in i for $\beta < 1$, and increasing in i for $\beta > 1$.

Theorem. *When weights are uniform, (i) for $\beta < 1$, HRB-CUM favors more popular item compared to HPB-CUM, i.e., $\exists j \in (1, n)$ s.t. $h_i^r > h_i^p$, , $\forall i < j$, and $h_i^r < h_i^p$, $\forall i > j$; and (i) for $\beta > 1$, HRB-CUM favors less popular item compared to HPB-CUM, i.e., $\exists l \in (1, n)$ s.t. $h_i^r < h_i^p$, $\forall i < l$, and $h_i^r > h_i^p$, $\forall i > l$. In particular, if $j, l \in \mathbb{Z}^+$, then $h_j^r = h_j^p$, and $h_l^r = h_l^p$.*

Proof. We first consider $\beta < 1$, i.e., h_i^r is decreasing in i. We have

$$\begin{aligned}h_1^r &= \frac{\mu_1^{\frac{1}{\beta}-1}}{\sum_{j=1}^n \mu_j^{\frac{1}{\beta}-1}} B > \frac{\mu_1^{\frac{1}{\beta}-1}}{n\mu_1^{\frac{1}{\beta}-1}} B = \frac{B}{n} = h_1^p, \\ h_n^r &= \frac{\mu_n^{\frac{1}{\beta}-1}}{\sum_{j=1}^n \mu_j^{\frac{1}{\beta}-1}} B < \frac{\mu_n^{\frac{1}{\beta}-1}}{n\mu_n^{\frac{1}{\beta}-1}} B = \frac{B}{n} = h_n^p.\end{aligned} \tag{43}$$

Since h_i^r is decreasing in i and $h_i^p = \frac{B}{n}$ for any $i = 1, \cdots, n$, thus, there must exist an intersection point $1 < j < n$ such that $h_j^r = h_j^p$, satisfying that $h_k^r > h_k^p$ for $k = 1, \cdots, j-1$ and $h_k^r < h_k^p$ for $k = j+1, \cdots, n$.

Therefore, when $\beta < 1$, we know that HRB-CUM favors more popular item compared to HPB-CUM.

Similarly, when $\beta > 1$, h_i^r is increasing in i. We have

$$\begin{aligned}h_1^r &= \frac{\mu_1^{\frac{1}{\beta}-1}}{\sum_{j=1}^n \mu_j^{\frac{1}{\beta}-1}} B < \frac{\mu_1^{\frac{1}{\beta}-1}}{n\mu_1^{\frac{1}{\beta}-1}} B = \frac{B}{n} = h_1^p, \\ h_n^r &= \frac{\mu_n^{\frac{1}{\beta}-1}}{\sum_{j=1}^n \mu_j^{\frac{1}{\beta}-1}} B > \frac{\mu_n^{\frac{1}{\beta}-1}}{n\mu_n^{\frac{1}{\beta}-1}} B = \frac{B}{n} = h_n^p.\end{aligned} \tag{44}$$

Again, as h_i^r is increasing in i and $h_i^p = \frac{B}{n}$ for any $i = 1, \cdots, n$, thus, there must exist an intersection point $1 < l < n$ such that $h_i^r = h_i^p$, satisfying that $h_k^r < h_k^p$ for $k = 1, \cdots, l-1$ and $h_k^r > h_k^p$ for $k = l+1, \cdots, n$.

Therefore, when $\beta > 1$, we know that HRB-CUM favors less popular item compared to HPB-CUM. □

Proof for Theorem 4:

Proof. Above theorem can be proved in a similar manner to that of uniform distribution. □

Proof for Theorem 3:

Proof. Above theorem can be proved in a similar manner to that of uniform distribution. □

D. Decentralized Algorithms in Section VI

1) Non-linear Equations for Dual in HRB-CUM: We show the existence of a solution of (28).

Theorem. *For any $\eta^{(k)}, w_i > 0$ and $0 \le k_i \le 1$, there always exists a unique solution in $[0, 1]$ for*

$$e(h_i^{(k)}) = \mu_i^{1-\beta} \frac{w_i\big(1 - h_i^{(k)}\big)^{k_i}}{\eta^{(k)}(1 - k_i)} - \big(h_i^{(k)}\big)^{\beta} = 0. \tag{45}$$

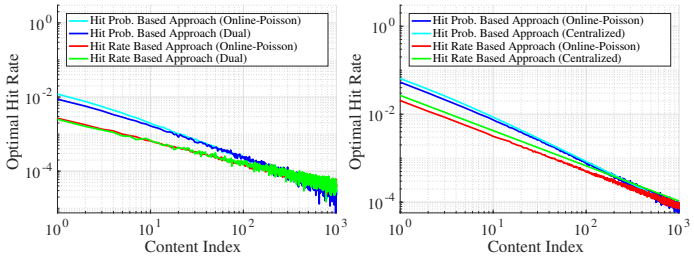

Fig. 6: Poisson online approximation to Hyperexponential *(Left)* and weibull *(Right)* inter-arrivals

Proof. For $h_i^{(k)} = 0$ and $h_i^{(k)} = 1$, we have

$$e(0) = \frac{\mu_i^{1-\beta} w_i}{\eta^{(k)}(1-k_i)}, \quad e(1) = -1.$$

Furthermore,

$$e'(h_i^{(k)}) = -\frac{\mu_i^{1-\beta} w_i k_i (1-h_i^{(k)})^{k_i-1}}{\eta^{(k)}(1-k_i)} - \beta (h_i^{(k)})^{\beta-1} < 0, \quad \forall h_i^{(k)} \in [0,1]. \quad (46)$$

Thus $e(\cdot)$ is decreasing in $h_i^{(k)}$. Since $h_i^{(k)} \in [0,1]$, $e(0) > 0$ and $e(1) < 0$, therefore, there always exists a unique solution to (45) in $[0,1]$. □

E. Poisson Approximation for various distributions

1) Hyperexponential Distribution: W.l.o.g., we consider the arrivals follow a hyperexponential distribution with phase probabilities $p_{1i} = p_{2i} = 0.5$, and phase rate parameters θ_1 and θ_2 Zipf distributed with rates 0.4 and 0.8, respectively. From Figure 4 (b), we can see that the optimal hit rates obtained through (30) exactly match those obtained from Dual under hyperexponential distribution by solving fixed point equation in (29). Similar performance was obtained for other parameters, especially for $p_{1i} \ll p_{2i}$, hence are omitted here.

2) Weibull Distribution: Similarly, we consider a Weibull distribution with shape parameter $k_i = 0.5$. From Figure 4 (c), it is clear that this approximation is accurate. Note that when $k_i \to 1$, Weibull behaves more closed to exponential distribution, hence the accuracy of this approximation can be further improved. For smaller value of k_i, it has been shown that Weibull can be well approximated by hyperexponential distribution [19]. The performance of Poisson approximation to hyperexponential distribution has been discussed in Section X-E1.

F. Limiting Behavior of 2-MMPP

***Case** 1**: When** $r_{1i} \to \infty$ **and** $r_{2i} \to \infty$**, i.e.,** $x_i \to \infty$**:** From Equation (20), we get*

$$u_{1i} = \frac{1}{2}\left[\theta_{1i} + \theta_{2i} + r_{1i} + r_{2i} - \sqrt{(\theta_{1i} - \theta_{2i} + r_{1i} - r_{2i})^2 + 4r_{1i}r_{2i}}\right]$$
$$= \frac{1}{2}\left[\frac{(\theta_{1i} + \theta_{2i} + r_{1i} + r_{2i})^2 - (\theta_{1i} - \theta_{2i} + r_{1i} - r_{2i})^2 - 4r_{1i}r_{2i}}{\theta_{1i} + \theta_{2i} + r_{1i} + r_{2i} + \sqrt{(\theta_{1i} - \theta_{2i} + r_{1i} - r_{2i})^2 + 4r_{1i}r_{2i}}}\right]$$
$$= \frac{2\theta_{1i}\theta_{2i} + 2\theta_{1i}a_{2i}x_i + 2\theta_{2i}a_{1i}x_i}{\theta_{1i} + \theta_{2i} + (a_{1i} + a_{2i})x_i + \sqrt{(\theta_{1i} - \theta_{2i} + (a_{1i} - a_{2i})x_i)^2 + 4a_{1i}a_{2i}x_i^2}}. \quad (47)$$

When $x_i \to \infty$, by applying L'Hospital's rule, we have

$$u_{1i} = \frac{2\theta_{1i}a_{2i} + 2\theta_{2i}a_{1i}}{a_{1i} + a_{2i} + \lim_{x_i\to\infty} \frac{(\theta_{1i} - \theta_{2i} + (a_{1i} - a_{2i})x_i)(a_{1i} - a_{2i}) + 4a_{1i}a_{2i}x_i}{\sqrt{(\theta_{1i} - \theta_{2i} + (a_{1i} - a_{2i})x_i)^2 + 4a_{1i}a_{2i}x_i^2}}}$$
$$= \frac{2\theta_{1i}a_{2i} + 2\theta_{2i}a_{1i}}{a_{1i} + a_{2i} + \lim_{x_i\to\infty} \frac{(\theta_{1i} - \theta_{2i})(a_{1i} - a_{2i}) + (a_{1i} + a_{2i})^2 x_i}{\sqrt{(\theta_{1i} - \theta_{2i})^2 + 2(\theta_{1i} - \theta_{2i})(a_{1i} - a_{2i})x_i + (a_{1i} + a_{2i})^2 x_i^2}}}$$
$$= \frac{2\theta_{1i}a_{2i} + 2\theta_{2i}a_{1i}}{a_{1i} + a_{2i} + \lim_{x_i\to\infty} \frac{(a_{1i} + a_{2i})^2 x_i}{(a_{1i} + a_{2i})x_i}}$$
$$= \frac{2\theta_{1i}a_{2i} + 2\theta_{2i}a_{1i}}{a_{1i} + a_{2i} + a_{1i} + a_{2i}}$$
$$= \frac{\theta_{1i}a_{2i} + \theta_{2i}a_{1i}}{a_{1i} + a_{2i}}. \quad (48)$$

Similarly, we obtain $u_{2i} = \infty$.

Again, from Equation (20),

$$q_{1i} = \frac{\theta_{2i}^2 r_{1i} + \theta_{1i}^2 r_{2i}}{(\theta_{1i}r_{2i} + \theta_{2i}r_{1i})(u_{1i} - u_{2i})} - \frac{u_{2i}}{u_{1i} - u_{2i}}$$
$$= \frac{\theta_{2i}^2 a_{1i} + \theta_{1i}^2 a_{2i}}{(\theta_{1i}a_{2i} + \theta_{2i}a_{1i})(-\delta)} + \frac{\theta_{1i} + \theta_{2i} + (a_{1i} + a_{2i})x_i}{2\delta} + \frac{1}{2}$$
$$= \frac{(\theta_{1i}a_{2i} + \theta_{2i}a_{1i})(a_{1i} + a_{2i})x + \theta_{1i}\theta_{2i}(a_{1i} + a_{2i}) - \theta_{1i}^2 a_{2i} - \theta_{2i}^2 a_{1i}}{2(\theta_{1i}a_{2i} + \theta_{2i}a_{1i})\sqrt{(\theta_{1i} - \theta_{2i} + (a_{1i} - a_{2i})x_i)^2 + 4a_{1i}a_{2i}x_i^2}}$$
$$+ \frac{1}{2}, \quad (49)$$

when $x_i \to \infty$, by applying L'Hospital's rule, we have q_{1i}

$$= \frac{1}{2} + \frac{(\theta_{1i}a_{2i} + \theta_{2i}a_{1i})(a_{1i} + a_{2i})}{2(\theta_{1i}a_{2i} + \theta_{2i}a_{1i}) \lim_{x_i\to\infty} \frac{(\theta_{1i} - \theta_{2i} + (a_{1i} - a_{2i})x_i)(a_{1i} - a_{2i}) + 4a_{1i}a_{2i}x_i}{\sqrt{(\theta_{1i} - \theta_{2i} + (a_{1i} - a_{2i})x_i)^2 + 4a_{1i}a_{2i}x_i^2}}}$$
$$= \frac{1}{2} + \frac{(\theta_{1i}a_{2i} + \theta_{2i}a_{1i})(a_{1i} + a_{2i})}{2(\theta_{1i}a_{2i} + \theta_{2i}a_{1i})(a_{1i} + a_{2i})}$$
$$= \frac{1}{2} + \frac{1}{2} = 1, \quad (50)$$

thus $q_{2i} = 0$.

***Case** 2**: When** $r_{1i} \to 0$ **and** $r_{2i} \to 0$**, i.e.,** $x_i \to 0$**:** W.l.o.g., we assume $\theta_{1i} \geq \theta_{2i}$. From Equation (20),*

$$\delta = \theta_{1i} - \theta_{2i}, \quad (51)$$

then

$$u_{1i} = \frac{1}{2}[\theta_{1i} + \theta_{2i} + (a_{1i} + a_{2i})x_i - \delta] = \theta_{2i}, \quad (52)$$

similarly, we have $u_{2i} = \theta_{1i}$.

Again, from Equation (20), when $x_i \to 0$, we obtain q_{1i}

$$= \frac{1}{2} + \frac{(\theta_{1i}a_{2i} + \theta_{2i}a_{1i})(a_{1i} + a_{2i})x + \theta_{1i}\theta_{2i}(a_{1i} + a_{2i}) - \theta_{1i}^2 a_{2i} - \theta_{2i}^2 a_{1i}}{2(\theta_{1i}a_{2i} + \theta_{2i}a_{1i})\delta}$$
$$= \frac{1}{2} + \frac{\theta_{1i}\theta_{2i}(a_{1i} + a_{2i}) - \theta_{1i}^2 a_{2i} - \theta_{2i}^2 a_{1i}}{2(\theta_{1i}a_{2i} + \theta_{2i}a_{1i})(\theta_{1i} - \theta_{2i})}$$
$$= \frac{\theta_{2i}a_{1i}}{\theta_{1i}a_{2i} + \theta_{2i}a_{1i}}, \quad (53)$$

then $q_{2i} = \frac{\theta_{1i}a_{2i}}{\theta_{1i}a_{2i} + \theta_{2i}a_{1i}}$.

REFERENCES

[1] O. I. Aven, E. G. Coffman, and Y. A. Kogan. *Stochastic Analysis of Computer Storage*. Springer Science & Business Media, 1987.

[2] F. Baccelli and P. Brémaud. *Elements of Queueing Theory: Palm Martingale Calculus and Stochastic Recurrences*, volume 26. Springer Science & Business Media, 2013.

[3] D. Berger, P. Gland, S. Singla, and F. Ciucu. Exact Analysis of TTL Cache Networks. *Performance Evaluation*, 79:2–23, 2014.

[4] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. In *ACM IMC*, 2007.

[5] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. Analyzing the Video Popularity Characteristics of Large-Scale User Generated Content Systems. *IEEE/ACM Transactions on Networking*, 17(5):1357–1370, 2009.

[6] H. Che, Y. Tung, and Z. Wang. Hierarchical Web Caching Systems: Modeling, Design and Experimental Results. *IEEE Journal on Selected Areas in Communications*, 20(7):1305–1314, 2002.

[7] M. Dehghan, L. Massoulie, D. Towsley, D. Menasche, and Y. Tay. A Utility Optimization Approach to Network Cache Design. In *IEEE INFOCOM*, 2016.

[8] R. Fagin. Asymptotic Miss Ratios over Independent References. *Journal of Computer and System Sciences*, 14(2):222–250, 1977.

[9] A. Feldmann and W. Whitt. Fitting Mixtures of Exponentials to Long-tail Distributions to Analyze Network Performance Models. In *IEEE INFOCOM*, 1997.

[10] A. Ferragut, I. Rodríguez, and F. Paganini. Optimizing TTL Caches under Heavy-tailed Demands. In *ACM SIGMETRICS*, 2016.

[11] W. Fischer and K. Meier-Hellstern. The Markov-modulated Poisson Process (MMPP) Cookbook. *Performance Evaluation*, 18:149–171, 1992.

[12] N. C. Fofack, M. Dehghan, D. Towsley, M. Badov, and D. L. Goeckel. On the Performance of General Cache Networks. In *VALUETOOLS*, 2014.

[13] N. C. Fofack, P. Nain, G. Neglia, and D. Towsley. Analysis of TTL-based Cache Networks. In *VALUETOOLS*, 2012.

[14] N. C. Fofack, P. Nain, G. Neglia, and D. Towsley. Performance Evaluation of Hierarchical TTL-based Cache Networks. *Computer Networks*, 2014.

[15] C. Fricker, P. Robert, J. Roberts, and N. Sbihi. Impact of Traffic Mix on Caching Performance in a Content-Centric Network. In *INFOCOM WKSHPS*, 2012.

[16] M. Garetto, E. Leonardi, and v. Martina. A Unified Approach to the Performance Analysis of Caching Systems. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, 1(3):12, 2016.

[17] N. Gast and B. Van Houdt. Asymptotically Exact TTL-Approximations of the Cache Replacement Algorithms LRU(m) and h-LRU. In *ITC 28*, 2016.

[18] B. Jiang, P. Nain, and D. Towsley. On the Convergence of the TTL Approximation for an LRU Cache under Independent Stationary Rrequest Processes. *Arxiv preprint arXiv:1707.06204*, 2017.

[19] T. Jin and L. Gonigunta. Exponential Approximation to Weibull Renewal with Decreasing Failure Rate. *J. Stat. Comput. Simul.*, 80(3):273–285, 2010.

[20] J. Jung, A. Berger, and H. Balakrishnan. Analysis of TTL-based Cache Networks. In *IEEE INFOCOM*, 2003.

[21] S. Kang and D. Sung. Two-state MMPP Modelling of ATM Superposed Traffic Streams Based on The Characterisation of Correlated Interarrival Times. In *IEEE GLOBECOM*, 1995.

[22] F. Kelly. Charging and Rate Control for Elastic Traffic. *Transactions on Emerging Telecommunications Technologies*, 8(1):33–37, 1997.

[23] F. P. Kelly, A. K. Maulloo, and D. K. Tan. Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Journal of the Operational Research society*, 49(3):237–252, 1998.

[24] J. Li, S. Shakkottai, J. C. S. Lui, and V. Subramanian. Accurate Learning or Fast Mixing? Dynamic Adaptability of Caching Algorithms. *IEEE Journal on Selected Areas in Communications*, 2018.

[25] R. Ma and D. Towsley. Cashing in on Caching: On-demand Contract Design with Linear Pricing. In *CoNext*, 2015.

[26] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge University Press, 2009.

[27] N. K. Panigrahy, J. Li, and D. Towsley. Hit rate vs. hit probability based cache utility maximization. In *ACM MAMA*, 2017.

[28] N. K. Panigrahy, J. Li, and D. Towsley. Network Cache Design under Stationary Requests: Exact Analysis and Poisson Approximation. *Arxiv preprint arXiv:1712.07307*, 2017.

[29] N. K. Panigrahy, J. Li, F. Zafari, D. Towsley, and P. Yu. Optimizing Timer-based Policies for General Cache Networks. *Arxiv preprint arXiv:1711.03941*, 2017.

[30] V. Paxson and S. Floyd. Wide-Area Traffic: The Failure of Poisson Modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.

[31] P. Rodriguez, C. Spanner, and E. W. Biersack. Analysis of Web Caching Architectures: Hierarchical and Distributed Caching. *IEEE/ACM Transactions on Networking*, 2001.

[32] A. E. Smith and D. W. Coit. *Evolutionary Computation*. Institute of Physics Publishing and Cambridge University Press, 1996.

[33] R. Srikant and L. Ying. *Communication Networks: an Optimization, Control, and Stochastic Networks Perspective*. Cambridge University Press, 2013.

[34] G. Weinberg. Kullback Leibler Divergence and the Pareto Exponential Approximation. *SpringerPlus 5*, 2016.

[35] M. Zink, K. Suh, Y. Gu, and J. Kurose. Watch Global, Cache Local: YouTube Network Traffic at a Campus Network: Measurements and Implications. In *Electronic Imaging*, 2008.