

Asymptotic Optimality of the Static Frequency Caching in the Presence of Correlated Requests

PREDRAG R. JELENKOVIĆ *

Department of Electrical Engineering
Columbia University, New York

ANA RADOVANOVIĆ †

Google, Inc.
New York

October 2007; revised August 2008; finalized March 2009

Abstract

It is well known that the static caching algorithm that keeps the most frequently requested documents in the cache is optimal in case when documents are of the same size and requests are independent and equally distributed. However, it is hard to develop explicit and provably optimal caching algorithms when requests are statistically correlated. In this paper, we show that keeping the most frequently requested documents in the cache is still optimal for large cache sizes even if the requests are strongly correlated.

Keywords: Web caching, cache fault probability, average-case analysis, least-frequently-used caching, least-recently-used caching, long-range dependence

1 Introduction

One of the important problems facing current and future network designs is the ability to store and efficiently deliver a huge amount of multimedia information in a timely manner. Web caching is widely recognized as an effective solution that improves the efficiency and scalability of multimedia content delivery, benefits of which have been repeatedly verified in practice. For an introduction to the concept of Web caching, the most recent tutorials, references and the latest technology, an interested reader is referred to the Web caching and content delivery Web page [8].

Caching is essentially a process of storing information closer to users so that Internet service providers, delivering a given content, do not have to go back to the origin servers every time the content is requested. It is clear that keeping more popular documents closer to the users can significantly reduce the traffic between the cache and the main servers and, therefore, improve the network performance, i.e., reduce the download latency and network congestion. One of the key components of engineering efficient Web caching systems is designing document placement/replacement algorithms (policies) that are managing cache content, i.e., selecting and possibly dynamically updating a collection of cached documents.

The main performance objective in creating and implementing these algorithms is minimizing the long-term fault probability, i.e., the average number of misses during a long time period. In the context of equal size documents and independent reference model, i.e., independent and identically distributed requests, it is well known (see [5], Chapter 6 of [15]) that keeping the most popular documents in the cache optimizes the long term cache performance; throughout this paper we refer to this algorithm as *static frequency caching*. A practical implementation of this algorithm is known as Least-Frequently-Used rule (LFU). However, the previous model does not incorporate any of the recently observed properties of the Web environment, such as: variability of document sizes, presence of temporal locality in the request patterns (e.g., see [9], [14], [2], [6], [7] and references therein), variability in document popularities (e.g., see [3]) and retrieval latency (e.g., see [1]).

Many heuristic algorithms that exploit the previously mentioned properties of the Web environment have been proposed, e.g., see [7], [5], [13] and references therein. However, there are no explicit algorithms that are provably optimal when the requests are statistically correlated even if documents are of equal size. Our main result of this paper, stated in Theorem 1 of Section 3, shows that, in the generality of semi-Markov modulated requests, the static frequency caching algorithm is still optimal for large cache sizes. The semi-Markov modulated processes, described

*Predrag Jelenković, Department of Electrical Engineering, Columbia University, New York, NY 10027, predrag@ee.columbia.edu.

†Corresponding author: Ana Radovanović, Google Inc., New York, NY 10011, anaradovanovic@google.com.

in Section 2, are capable of modeling a wide range of statistical correlation, including the long-range dependence (LRD) that was repeatedly experimentally observed in Web access patterns; these types of models were recently used in [10] and their potential confirmed on real Web traces in [9]. In Section 4, under mild additional assumptions, we show how our result extends to variable page sizes. Our optimality result provides a benchmark for evaluating other heuristic schemes, suggesting that any heuristic caching policy that approximates well the static frequency caching should achieve the nearly-optimal performance for large cache sizes. In particular, in conjunction with our result from [10], we show that a widely implemented Least-Recently-Used (LRU) caching heuristic is, for semi-Markov modulated requests and generalized Zipf's law document frequencies, asymptotically only a factor of 1.78 away from the optimal. Furthermore, similar results can be expected to hold for the improved version of the LRU caching, termed Persistent Access Caching, that was recently proposed and analyzed in [12].

2 Modeling statistical dependency in the request process

In this section we describe a semi-Markov modulated request process. As stated earlier, this model is capable of capturing a wide range of statistical correlation, including the commonly empirically observed LRD. This approach was recently used in [10], where one can find more details and examples.

Let a sequence of requests arrive at Poisson points $\{\tau_n, -\infty < n < \infty\}$ of unit rate. At each point τ_n , we use R_n , $R_n \in \{1, 2, \dots, N\}$, to denote a document that has been requested, i.e., the event $\{R_n = i\}$ represents a request for document i at time τ_n ; we assume that the sequence $\{R_n\}$ is independent of the arrival Poisson points $\{\tau_n\}$ and that $\mathbb{P}[R_n = i] > 0$ for all i and $\mathbb{P}[R_n < \infty] = 1$.

Next, we describe the dependency structure of the request sequence $\{R_n\}$. We consider the class of finite-state, stationary and ergodic semi-Markov processes J , with jumps at almost surely strictly increasing points $\{T_n, -\infty < n < \infty\}$, $T_0 \leq 0 < T_1$. Let process $\{J_{T_n}, -\infty < n < \infty\}$ be an irreducible Markov chain that is independent of $\{\tau_n\}$, has finitely many states $\{1, \dots, M\}$ and transition matrix $\{p_{ij}\}$. Then, we construct a piecewise constant and right-continuous *modulating process* $\{J_t\}$ such that

$$J_t = J_{T_n}, \quad \text{if } T_n \leq t < T_{n+1};$$

for more details on the construction of process J_t , $t \in \mathbb{R}$ see Subsection 4.3 of [10]. Let $\pi_r = \mathbb{P}[J_t = r]$, $1 \leq r \leq M$, be the stationary distribution of J and, to avoid trivialities, we assume that $\min_r \pi_r > 0$. For each $1 \leq r \leq M$, let $q_i^{(r)}$, $1 \leq i \leq N \leq \infty$, be a probability mass function, where $q_i^{(r)}$ is used to denote the probability of requesting item i when the underlying process J is in state r . Next, the probability law of $\{R_n\}$ is uniquely determined by the modulating process J according to the following conditional distribution,

$$\mathbb{P}[R_l = i_l, 1 \leq l \leq n | J_t, 0 \leq t \leq \tau_n] = \prod_{l=1}^n q_{i_l}^{(J_{\tau_l})}, \quad n \geq 1, \quad (1)$$

i.e., the sequence of requests R_n is conditionally independent given the modulating process J . Given the properties introduced above, it is easy to conclude that the constructed request process $\{R_n\}$ is stationary and ergodic as well. We will use

$$q_i = \mathbb{P}[R_n = i] = \sum_{r=1}^M \pi_r q_i^{(r)}$$

to express the marginal request distribution, with the assumption that $q_i > 0$ for all $i \geq 1$. In addition, assume that requests are enumerated according to the non-increasing order of marginal request popularities, i.e., $q_1 \geq q_2 \geq \dots$.

In this paper we are using the following standard notation. For any two real functions $a(t)$ and $b(t)$ and fixed $t_0 \in \mathbb{R} \cup \{\infty\}$ we use $a(t) \sim b(t)$ as $t \rightarrow t_0$ to denote $\lim_{t \rightarrow t_0} [a(t)/b(t)] = 1$. Similarly, we say that $a(t) \gtrsim b(t)$ as $t \rightarrow t_0$ if $\liminf_{t \rightarrow t_0} a(t)/b(t) \geq 1$; $a(t) \lesssim b(t)$ has a complementary definition, i.e., $\limsup_{t \rightarrow t_0} a(t)/b(t) \leq 1$.

Throughout the paper we will exploit the renewal (regenerative) structure of the semi-Markov process. In this regard, let $\{\mathcal{T}_i\}$, $T_0 \leq 0 < T_1$, be a subset of points $\{T_n\}$ for which $J_{T_n} = 1$. Then, it is well known that $\{\mathcal{T}_i\}$ is a renewal process and that sets of variables $\{J_t, \mathcal{T}_j \leq t < \mathcal{T}_{j+1}\}$ are independent for different j and identically distributed, i.e., $\{\mathcal{T}_i\}$ are regenerative points for $\{J_t\}$. Furthermore, the conditional independence of $\{R_n\}$ given $\{J_t\}$, implies that $\{\mathcal{T}_i\}$ are regenerative points for R_n as well.

Next we define $\mathcal{R}_r(u, t)$, $1 \leq r \leq M$, to be a set of distinct requests that arrived in interval $[u, t)$, $u \leq t$, and denote by $N_r(u, t)$, $1 \leq r \leq M$, the number of requests in interval $[u, t)$ when process J_i is in state r . Furthermore, let $N(u, t) \triangleq N_1(u, t) + \dots + N_M(u, t)$ represent the total number of requests in $[u, t)$; note that $N(u, t)$ has Poisson distribution with mean $t - u$.

The following technical lemma will be used in the proof of the main result of this paper.

Lemma 1 *For the request process introduced above, the following asymptotic relation holds*

$$\mathbb{P}[i \in \mathcal{R}(\mathcal{T}_1, \mathcal{T}_2)] \sim q_i \mathbb{E}[\mathcal{T}_2 - \mathcal{T}_1] \text{ as } i \rightarrow \infty, \quad (2)$$

where $\mathcal{R}(u, t) \triangleq \mathcal{R}_1(u, t) \cup \dots \cup \mathcal{R}_M(u, t)$.

Proof: Given in Section 5. ◇

3 Caching policies and the optimality

Consider infinitely many documents of unit size out of which x can be stored in a local memory referred to as cache. When an item is requested, the cache is searched first and we say that there is a cache hit if the item is found in the cache. In this case the cache content is left unchanged. Otherwise, we say that there is a cache fault/miss and the missing item is brought in from the outside world. At the time of a fault, a decision whether to replace some item from the cache with a missing item has to be made. We assume that replacements are optional, i.e., the cache content can be left unchanged even in the case of fault. A caching algorithm represents a set of document replacement rules.

We consider a class of caching algorithms whose information decisions are made using only the information of past and present requests and past decisions. More formally, let $\mathcal{C}_t^\pi \equiv \mathcal{C}_t^\pi(x)$ be a cache content at time t under policy π . When the request for a document R_n is made, the cache with content $\mathcal{C}_{\tau_n}^\pi$ is searched first. If document R_n is already in the cache ($R_n \in \mathcal{C}_{\tau_n}^\pi$), then we use the convention that no document is replaced. On the other hand, if document R_n is not an element of $\mathcal{C}_{\tau_n}^\pi$, then a document to be replaced is chosen from a set $\mathcal{C}_{\tau_n}^\pi \cup \{R_n\}$ using a particular eviction policy. At any moment of request, τ_n , the decision what to replace in the cache is based on $R_1, R_2, \dots, R_n, \mathcal{C}_{\tau_0}^\pi, \mathcal{C}_{\tau_1}^\pi, \dots, \mathcal{C}_{\tau_n}^\pi$. Note that this information already contains all the replacement decisions made up to time τ_n . This is the same information as the one used in the Markov decision framework [5].

The set of the previously described cache replacement policies, say \mathcal{P}_c , is quite large and contains mandatory caching rules (more typical for a computer memory environment), i.e., those rules that require replacements in the case of cache faults. Furthermore, the set \mathcal{P}_c also contains the static algorithm that places a fixed collection of documents $\mathcal{C}_t^\pi \equiv \mathcal{C}$ in the cache and then keeps the same content without ever changing it.

Now, define the long-run cache fault probability corresponding to the policy $\pi \in \mathcal{P}_c$ and a cache of size x as

$$P(\pi, x) \triangleq \limsup_{T \rightarrow \infty} \frac{\mathbb{E} \left[\sum_{\tau_n \in [0, T]} 1[R_n \notin \mathcal{C}_{\tau_n}^\pi] \right]}{T}, \quad (3)$$

recall that $\mathbb{E}N(0, T) = T$. Note that we use the lim sup in this definition since the limit may not exist in general and that, as defined before, $\mathcal{C}_{\tau_n}^\pi \equiv \mathcal{C}_{\tau_n}^\pi(x)$ is a function of x and we suppress it from the notation.

Next, we show that

$$P(\pi, x) = \limsup_{k \rightarrow \infty} \frac{\mathbb{E} \left[\sum_{\tau_n \in [0, \mathcal{T}_k]} 1[R_n \notin \mathcal{C}_{\tau_n}^\pi] \right]}{\mathbb{E}N(0, \mathcal{T}_k)}, \quad (4)$$

where \mathcal{T}_k are the regenerative points, as defined in the previous section. Note that estimating the previous expression is not straight forward since replacement decision depends on all previous requests, i.e., it depends on the past beyond the last regenerative point. To this end, for the lower bound, for any $0 < \epsilon < 1$, let $k \equiv k(T, \epsilon) \triangleq \lfloor T(1 - \epsilon)/\mathbb{E}[\mathcal{T}_2 - \mathcal{T}_1] \rfloor$, where $\lfloor u \rfloor$ is the largest integer that is less or equal to u . Then, note that

$$\begin{aligned} \frac{1}{T} \mathbb{E} \left[\sum_{\tau_n \in [0, T]} 1[R_n \notin \mathcal{C}_{\tau_n}^\pi] \right] &\geq \mathbb{E} \left[1[\mathcal{T}_k < T] \frac{\sum_{\tau_n \in [0, \mathcal{T}_k]} 1[R_n \notin \mathcal{C}_{\tau_n}^\pi]}{T} \right] \\ &\geq \mathbb{E} \left[\frac{\sum_{\tau_n \in [0, \mathcal{T}_k]} 1[R_n \notin \mathcal{C}_{\tau_n}^\pi]}{T} \right] - \mathbb{E} \left[1[\mathcal{T}_k > T] \frac{N(0, T)}{T} \right]. \end{aligned} \quad (5)$$

Next, using the Weak Law of Large Numbers for $\mathbb{P}[\mathcal{T}_k > T] \rightarrow 0$ (as $T \rightarrow \infty$) and the fact that $N(0, T)$ is Poisson with mean T in the preceding inequality, we obtain

$$P(\pi, x) \geq (1 - \epsilon) \limsup_{\substack{T \rightarrow \infty \\ k = \lfloor \frac{T(1-\epsilon)}{\mathbb{E}[\mathcal{T}_2 - \mathcal{T}_1]} \rfloor}} \frac{\mathbb{E} \left[\sum_{\tau_n \in [0, \mathcal{T}_k]} 1[R_n \notin \mathcal{C}_{\tau_n}^\pi] \right]}{\mathbb{E}N(0, \mathcal{T}_k)} = (1 - \epsilon) \limsup_{k \rightarrow \infty} \frac{\mathbb{E} \left[\sum_{\tau_n \in [0, \mathcal{T}_k]} 1[R_n \notin \mathcal{C}_{\tau_n}^\pi] \right]}{\mathbb{E}N(0, \mathcal{T}_k)},$$

since the set $\{k : k = \lfloor T(1 - \epsilon)/\mathbb{E}[\mathcal{T}_2 - \mathcal{T}_1] \rfloor, T > 0\}$ covers all integers. We complete the proof of the lower bound by passing $\epsilon \rightarrow 0$. The upper bound uses similar arguments where, in this case, k is defined as $k \equiv k(T, \epsilon) \triangleq \lfloor T(1 + \epsilon)/\mathbb{E}[\mathcal{T}_2 - \mathcal{T}_1] \rfloor$, and $P(\pi, x)$ is upper bounded as

$$\frac{1}{T} \mathbb{E} \left[\sum_{\tau_n \in [0, T]} 1[R_n \notin \mathcal{C}_{\tau_n}^\pi] \right] \leq \mathbb{E} \left[1[T < \mathcal{T}_k] \frac{\sum_{\tau_n \in [0, \mathcal{T}_k]} 1[R_n \notin \mathcal{C}_{\tau_n}^\pi]}{T} \right] + \mathbb{E} \left[1[T > \mathcal{T}_k] \frac{N(0, T)}{T} \right].$$

Then, similarly to earlier arguments, we derive the corresponding upper bound for $P(\pi, x)$ in (4).

Next, observe the static policy s , where $\mathcal{C}_{\tau_n}^\pi \equiv \{1, 2, \dots, x\}$ for every n . Then, due to the ergodicity of the request process, the long-run cache fault probability of the static policy is

$$P_s(x) \triangleq P(s, x) = \sum_{i > x} q_i.$$

Since the static policy belongs to the set of caching algorithms \mathcal{P}_c , we conclude that

$$P_s(x) \geq \inf_{\pi \in \mathcal{P}_c} P(\pi, x). \quad (6)$$

Our goal in this paper is to show that for large cache sizes x there is no caching policy that performs better, i.e., achieves long-term fault probability smaller than $P_s(x)$. This is stated in the following main result of this paper.

Theorem 1 *For the semi-Markov modulated request process defined in Section 2, the static policy that stores documents with the largest marginal popularities minimizes the long-term cache fault probability for large caches, i.e.,*

$$\inf_{\pi \in \mathcal{P}_c} P(\pi, x) \sim P_s(x) \text{ as } x \rightarrow \infty. \quad (7)$$

Remarks: (i) From the examination of the following proof it is clear that the result holds for any regenerative request process that satisfies Lemma 1. (ii) Though asymptotically long-term optimal, the static frequency rule possesses other undesirable properties such as high complexity and lack of adaptability to variations in the request patterns. However, its optimal performance presents an important benchmark for evaluating and comparing widely implemented caching policies in the Web environment. On the other hand, it is a question whether a widely accepted analysis of the cache miss ratio is the most relevant performance measure to analyze. A strong argument in support to this choice is that other measures would be harder (sometimes impossible) to analyze. However, in Section 4, we present some possible extensions of our results to the analysis of other objective functions, such as long-run average delay of fetching documents not found in the cache, or long-run average cost of retrieving documents outside of the cache, etc. (iii) Note that the condition $q_i, i \geq 1$, given in the previous section makes the problem of proving asymptotic optimality nontrivial. In case $q_i > 0$ for just a finite number of i 's, the document population would be finite and the result above would be trivially true. (iv) The preliminary version of this work was presented in the Workshop on Analytic Algorithms and Combinatorics (ANALCO'2006), Miami, Florida, January 2006.

Proof: In view of (6), we only need to show that $\inf_{\pi \in \mathcal{P}_c} P(\pi, x) \gtrsim P_s(x)$ as $x \rightarrow \infty$.

For any set \mathcal{A} , let $|\mathcal{A}|$ denote the number of elements in \mathcal{A} and $\mathcal{A} \setminus \mathcal{B}$ represent the set difference. Then, it is easy to see that the number of cache faults in $[t, u)$, $t < u$, is lower bounded by $|\mathcal{R}(t, u) \setminus \mathcal{C}_t^\pi|$ since every item that was not in the cache at time t results in at least one fault when requested for the first time; in particular, if $t = \mathcal{T}_j$, $u = \mathcal{T}_{j+1}$,

$$\sum_{\tau_n \in [\mathcal{T}_j, \mathcal{T}_{j+1})} 1[R_n \notin \mathcal{C}_{\tau_n}^\pi] \geq |\mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1}) \setminus \mathcal{C}_{\mathcal{T}_j}^\pi|. \quad (8)$$

This inequality and (4) results in

$$P(\pi, x) \geq \limsup_{k \rightarrow \infty} \frac{1}{\mathbb{E}N(0, \mathcal{T}_k)} \sum_{j=1}^{k-1} \mathbb{E}[|\mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1}) \setminus \mathcal{C}_{\mathcal{T}_j}^\pi|]. \quad (9)$$

Now, since we consider caching policies where replacement decisions depend only on the previous cache contents and requests, due to the renewal structure of the request process we conclude that for every $j \geq 1$ and all $i \geq 1$, events $\{i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})\}$ and $\{i \in \mathcal{C}_{\mathcal{T}_j}^\pi\}$ are independent and, therefore, for every $j \geq 1$,

$$\begin{aligned} \mathbb{E} \left[|\mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1}) \setminus \mathcal{C}_{\mathcal{T}_j}^\pi | 1[\mathcal{C}_{\mathcal{T}_j}^\pi = \mathcal{C}] \right] &= \sum_{i \geq 1} \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1}), i \notin \mathcal{C}] \mathbb{P}[\mathcal{C}_{\mathcal{T}_j}^\pi = \mathcal{C}] \\ &= \mathbb{P}[\mathcal{C}_{\mathcal{T}_j}^\pi = \mathcal{C}] \sum_{i \geq 1} \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})] 1[i \notin \mathcal{C}] \\ &\geq \mathbb{P}[\mathcal{C}_{\mathcal{T}_j}^\pi = \mathcal{C}] \inf_{\mathcal{C}: |\mathcal{C}|=x} \sum_{i \notin \mathcal{C}} \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})]. \end{aligned}$$

Then, after summing over all values of \mathcal{C} , for any $j \geq 1$ we obtain

$$\mathbb{E}[|\mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1}) \setminus \mathcal{C}_{\mathcal{T}_j}^\pi|] \geq \inf_{\mathcal{C}: |\mathcal{C}|=x} \sum_{i \notin \mathcal{C}} \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})]. \quad (10)$$

Next, we show that the cache content $\mathcal{C} = [1, x] \triangleq \{1, \dots, x\}$ achieves the infimum in the previous expression for large cache sizes. This is equivalent to proving that, as $x \rightarrow \infty$,

$$\inf_{\mathcal{C}: |\mathcal{C}|=x} \sum_{i \notin \mathcal{C}} \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})] \gtrsim \sum_{i \notin [1, x]} \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})]. \quad (11)$$

We will justify the previous statement by showing that for any set \mathcal{C} obtained from $[1, x]$ by placing documents from the set $\{x+1, \dots\}$ instead of those in $[1, x]$ can not result in $\sum_{i \notin \mathcal{C}} \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})] < (1 - \epsilon) \sum_{i \notin [1, x]} \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})]$ for large cache sizes x and any $0 < \epsilon < 1$.

Lemma 1 implies that for an arbitrarily chosen $\epsilon > 0$ there exists finite integer i_0 such that for all $i \geq i_0$

$$(1 - \epsilon)q_i \mathbb{E}[\mathcal{T}_{j+1} - \mathcal{T}_j] < \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})] < (1 + \epsilon)q_i \mathbb{E}[\mathcal{T}_{j+1} - \mathcal{T}_j]. \quad (12)$$

Thus, using the previous expression and $q_i \downarrow 0$ as $i \rightarrow \infty$, we conclude that for all $k \leq i_0$ there exists $x_0 \geq i_0$, such that for all $i \geq x_0$

$$\min_{1 \leq k \leq i_0} \mathbb{P}[k \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})] > \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})]. \quad (13)$$

Now, assume that the cache is of size $x \geq x_0$ and observe different cache contents \mathcal{C} obtained from $[1, x]$ by replacing its documents with items from $\{x+1, x+2, \dots\}$. Next, using (13), we conclude that replacing documents enumerated with $\{1, \dots, i_0\}$ can only increase the sum on the left-hand side of (11). On the other hand, observe cache contents \mathcal{C} that are obtained from $[1, x]$ by replacing documents enumerated as $\{i_0+1, \dots, x\}$ with items from $\{x+1, \dots\}$. Then, it is easy to see that proving inequality (11) is equivalent to showing that $\sum_{i \in [i_0+1, x]} \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})] \geq (1 - \epsilon) \sum_{i \in \mathcal{C} \setminus [1, i_0]} \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})]$, for any $0 < \epsilon < 1$. Next, since for any $i \geq i_0$ inequalities (12) hold, we conclude

$$\frac{\sum_{i \in [i_0+1, x]} \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})]}{\sum_{i \in \mathcal{C} \setminus [1, i_0]} \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})]} \geq \frac{(1 - \epsilon) \sum_{i \in [i_0+1, x]} q_i}{(1 + \epsilon) \sum_{i \in \mathcal{C} \setminus [1, i_0]} q_i} \geq \frac{1 - \epsilon}{1 + \epsilon},$$

where the second inequality in the previous expression follows from the monotonicity of q_i s. Then, by passing $\epsilon \rightarrow 0$ we prove inequality (11).

Note that after applying the lower bound (11) in (10), in conjunction with (9), the renewal nature of the regenerative points and Lemma 1, we obtain that, as $x \rightarrow \infty$,

$$\inf_{\pi \in \mathcal{P}_c} P(\pi, x) \gtrsim \sum_{i \geq x} q_i, \quad (14)$$

which completes the proof of the theorem. \diamond

4 Further extensions and concluding remarks

In this paper we prove that the static frequency rule minimizes the long term fault probability in the presence of correlated requests for large cache sizes.

There are several generalizations of our results that are worth mentioning. First, the definition of the fault probability in (4) can be generalized by replacing terms $1[R_n \notin \mathcal{C}_{\tau_n}^\pi]$ with $f(R_n)1[R_n \notin \mathcal{C}_{\tau_n}^\pi]$, where $f(i)$ could represent the cost of retrieving document i , e.g., the delay of fetching item i not found in the cache. Assume that $0 < f(i) \leq K < \infty$ and let \mathcal{S} be a set of x items such that $q_i f(i) \geq q_j f(j)$ for all $i \in \mathcal{S}$ and $j \notin \mathcal{S}$. Then, the following result holds:

Theorem 2 *For the semi-Markov modulated request process defined in Section 2, the static caching policy $\mathcal{C} \equiv \mathcal{S}$ minimizes the long-run average cost function $f(\cdot)$ (e.g., delay) for documents not found in the cache.*

Sketch of the proof: The proof of this theorem follows completely analogous arguments to those used in the proof of Theorem 1, and, in order to avoid repetitions, we outline its basic steps.

Similarly as in (3), the long-run average cost for documents not found in the cache that corresponds to the caching policy $\pi \in \mathcal{P}_c$ is defined as

$$D(\pi, x) \triangleq \limsup_{T \rightarrow \infty} \frac{\mathbb{E} \left[\sum_{\tau_n \in [0, T]} f(R_n) 1[R_n \notin \mathcal{C}_{\tau_n}^\pi] \right]}{T}.$$

Then, by using similar arguments to (4) - (6) and $0 < f(i) \leq K < \infty$, $i \geq 1$, we obtain that the long-run average cost of the static policy $\mathcal{C}_{\tau_n} \equiv \mathcal{S}$, $n \geq 1$, for the cache with size x satisfies

$$D_s(x) = \sum_{i \notin \mathcal{S}} f(i) q_i \geq \inf_{\pi \in \mathcal{P}_c} D(\pi, x). \quad (15)$$

Next, in order to prove

$$D_s(x) \lesssim \inf_{\pi \in \mathcal{P}_c} D(\pi, x) \text{ as } x \rightarrow \infty, \quad (16)$$

similarly as in the proof of Theorem 1, we lower bound the number of cache misses, and, therefore, the average cost in every regenerative interval $[\mathcal{T}_j, \mathcal{T}_{j+1})$, $j \geq 1$, as

$$\sum_{\tau_n \in [\mathcal{T}_j, \mathcal{T}_{j+1})} f(R_n) 1[R_n \notin \mathcal{C}_{\tau_n}^\pi] \geq \sum_{i \geq 1} f(i) 1[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1}), i \notin \mathcal{C}_{\mathcal{T}_j}].$$

Next, since we consider caching policies whose replacement decisions depend only on the past cache contents and requests, due to the renewal structure of the request process, we conclude that for any $j \geq 1$,

$$\mathbb{E} \left[\sum_{\tau_n \in [\mathcal{T}_j, \mathcal{T}_{j+1})} f(R_n) 1[R_n \notin \mathcal{C}_{\tau_n}^\pi] 1[\mathcal{C}_{\mathcal{T}_j}^\pi = \mathcal{C}] \right] \geq \mathbb{P}[\mathcal{C}_{\mathcal{T}_j}^\pi = \mathcal{C}] \sum_{i \notin \mathcal{C}} f(i) \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})]$$

and, thus, similarly as in (10), we obtain

$$\mathbb{E} \left[\sum_{\tau_n \in [\mathcal{T}_j, \mathcal{T}_{j+1})} f(R_n) 1[R_n \notin \mathcal{C}_{\tau_n}^\pi] \right] \geq \inf_{\mathcal{C}: |\mathcal{C}|=x} \sum_{i \notin \mathcal{C}} f(i) \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})].$$

Now, given the previous observations, the asymptotic inequality (16) is proved using analogous arguments to those in (12) - (14). Note that in the context of this result, inequality (13) becomes

$$\min_{1 \leq k \leq i_0} f(k) \mathbb{P}[k \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})] > f(i) \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_j, \mathcal{T}_{j+1})]$$

for all $i \geq x_0$, and we have analogous asymptotic linearity as in (12) since $q_i \downarrow 0$ as $i \rightarrow \infty$ and $0 < f(i) \leq K < \infty$. Finally, as in (13) - (14), the rest of the proof is based on proving that no replacements of documents in the set \mathcal{S} can lead to smaller long-run average delays for large cache sizes x . Thus, the asymptotic bound (16) holds and, in conjunction with (15), completes the proof of the theorem. \diamond

In addition to the previous generalization, in the context of documents with different sizes, one can prove the following result:

Theorem 3 Assume that documents have different sizes and that they are enumerated according to the non-increasing order of q_i/s_i , i.e., $q_1/s_1 \geq q_2/s_2 \geq \dots$, where s_i is the size of document i and $s_i \in \{s_1, \dots, s_D\}$, where $s_1, \dots, s_D < \infty$ and $D < \infty$. Then, for the semi-Markov modulated request process defined in Section 2, if $\sum_{j>i} q_j \sim \sum_{j \geq i} q_j$ as $i \rightarrow \infty$, i.e., $\sum_{j>i} q_j$ is long-tailed, the static rule that places documents with the smallest index in the cache, subject to the constraint $\sum_i s_i \leq x$, is asymptotically optimal.

Proof: In light of the identical arguments to those used in the proof of Theorem 1, it is not hard to show that the static policy minimizes the long run average number of misses for large cache sizes. More specifically, the optimal long-run cache fault probability is of the form

$$P_s(x) = \sum_{i \in \mathcal{N}_x} q_i, \quad (17)$$

where \mathcal{N}_x is the set of document indices that minimizes (17) subject to the constraint $\sum_{i \notin \mathcal{N}_x} s_i \leq x$. The previous problem is a knapsack problem (see Section 5.2 of [16] for further explanations). It is shown that in the case where objects can be split to exactly fill the knapsack, the policy that minimizes (17) is the one that places documents with the largest q_i/s_i values in the cache until an object, say n_x th, fails to fit. Then, the optimal solution is to split document n_x to fill the cache completely. Since that is not possible in the case of document caching, it is not hard to see that the fault probability for the optimal static placement in our case is between the optimal fault probabilities in the case of cache sizes $\sum_{i=1}^{n_x-1} s_i$ and $\sum_{i=1}^{n_x} s_i$ (note that $\sum_{i=1}^{n_x-1} s_i \leq x < \sum_{i=1}^{n_x} s_i$). Now, since n_x monotonically increases as x increases, in conjunction with the long-tailed assumption of the theorem, $\sum_{i \geq n_x} q_i \sim \sum_{i > n_x} q_i$ as $x \rightarrow \infty$, we conclude the proof of the theorem. \diamond

Finally, in light of our recent result on the asymptotic performance of the ordinary LRU caching rule in the presence of semi-Markov modulated requests and Zipf's law marginal distributions ($q_i \sim c/i^\alpha$ as $i \rightarrow \infty$, $c > 0$) obtained in Theorem 3 of [10], asymptotic optimality of the static frequency rule implies that the LRU is factor $e^\gamma \approx 1.78$ away from the optimal (γ is the Euler constant, i.e. $\gamma \approx 0.57721 \dots$). Therefore, in view of other desirable properties, its self-organizing nature and low complexity, the LRU rule has excellent performance even in the presence of statistically correlated requests. Furthermore, stronger optimality conclusions could be drawn for the recently proposed versions of the LRU policy (see [11] and [12]), given that the performance analysis of these algorithms can be extended to the correlated setting such as the one in [10].

5 Proof of Lemma 1

In this section, we prove the asymptotic relation (2) stated at the end of Section 2.

Note that

$$\begin{aligned} \mathbb{P}[i \in \mathcal{R}(\mathcal{T}_1, \mathcal{T}_2)] &= 1 - \mathbb{P}[i \notin \mathcal{R}_1(\mathcal{T}_1, \mathcal{T}_2), \dots, i \notin \mathcal{R}_M(\mathcal{T}_1, \mathcal{T}_2)] \\ &= \mathbb{E} \left[1 - (1 - q_i^{(1)})^{N_1} \dots (1 - q_i^{(M)})^{N_M} \right], \end{aligned} \quad (18)$$

where $N_r \triangleq N_r(\mathcal{T}_1, \mathcal{T}_2)$, $1 \leq r \leq M$. Then, since $q_i \rightarrow 0$ as $i \rightarrow \infty$ and $\min_r \pi_r > 0$, it follows that $q_i^{(r)} \rightarrow 0$ as $i \rightarrow \infty$, $1 \leq r \leq M$. In addition, $1 - e^{-x} \leq x$ for all $x \geq 0$ and for any $1 > \epsilon > 0$, there exists $x_0(\epsilon) > 0$, such that for all $0 \leq x \leq x_0(\epsilon)$ inequality $1 - x \geq e^{-x(1+\epsilon)}$ holds, and, therefore, for i large enough

$$\begin{aligned} \mathbb{E} \left[1 - e^{-(q_i^{(1)} N_1 + \dots + q_i^{(M)} N_M)} \right] &\leq \mathbb{E} \left[1 - (1 - q_i^{(1)})^{N_1} \dots (1 - q_i^{(M)})^{N_M} \right] \\ &\leq \mathbb{E} \left[1 - e^{-(1+\epsilon)(q_i^{(1)} N_1 + \dots + q_i^{(M)} N_M)} \right]. \end{aligned} \quad (19)$$

Then, since $1 - e^{-x} \leq x$ for $x \geq 0$, we obtain, for i large enough,

$$\mathbb{E} \left[1 - e^{-(1+\epsilon)(N_1 q_i^{(1)} + \dots + N_M q_i^{(M)})} \right] \leq (1 + \epsilon) \mathbb{E} \left[q_i^{(1)} N_1 + \dots + q_i^{(M)} N_M \right]. \quad (20)$$

Next, let $N \triangleq N_1 + \dots + N_M$. Then, we show that $q_i^{(1)} \mathbb{E} N_1 + \dots + q_i^{(M)} \mathbb{E} N_M = q_i \mathbb{E} N$. From the ergodicity of J_t , it follows that

$$\mathbb{P}[J_t = r] = \frac{\mathbb{E} \mathcal{T}_{1r}}{\mathbb{E}[\mathcal{T}_2 - \mathcal{T}_1]},$$

where \mathcal{T}_{1r} , $1 \leq r \leq M$, is the length of time that J_t spends in state r during the renewal interval $(\mathcal{T}_1, \mathcal{T}_2)$ (see Section 1.6 of [4]). Finally, using $\mathbb{E}N = \mathbb{E}[\mathcal{T}_2 - \mathcal{T}_1]$ and $\mathbb{E}N_r = \mathbb{E}\mathcal{T}_{1r}$, $1 \leq r \leq M$ (Poisson process of rate 1), in conjunction with (20), we conclude, for i large

$$\mathbb{E} \left[1 - e^{-(1+\epsilon)(N_1 q_i^{(1)} + \dots + N_M q_i^{(M)})} \right] \leq (1 + \epsilon) q_i \mathbb{E}[\mathcal{T}_2 - \mathcal{T}_1]. \quad (21)$$

Next, we estimate the lower bound for the left hand side in (19). After conditioning, we obtain

$$\mathbb{E} \left[1 - e^{-(N_1 q_i^{(1)} + \dots + N_M q_i^{(M)})} \right] \geq \mathbb{E} \left[\left(1 - e^{-(N_1 q_i^{(1)} + \dots + N_M q_i^{(M)})} \right) 1 \left[N \leq \bar{q}_i^{-\frac{1}{2}} \right] \right], \quad (22)$$

where $q_i^{(r)} \leq \bar{q}_i \triangleq q_i / \min_r \pi_r \leq H q_i$, $1 \leq r \leq M$, and some large enough constant $0 < H < \infty$. Then, note that for every $\omega \in \{N \leq \bar{q}_i^{-\frac{1}{2}}\}$, $q_i^{(1)} N_1 + \dots + q_i^{(M)} N_M \leq \frac{\bar{q}_i}{\sqrt{\bar{q}_i}} = \sqrt{\bar{q}_i}$. In addition, for any $1 > \epsilon > 0$, there exists $x_\epsilon > 0$, such that for all $0 \leq x \leq x_\epsilon$ inequality $1 - e^{-x} \geq (1 - \epsilon)x$ holds and, therefore, for i large enough such that $\sqrt{\bar{q}_i} \leq x_\epsilon$

$$\begin{aligned} \mathbb{E} \left[\left(1 - e^{-(N_1 q_i^{(1)} + \dots + N_M q_i^{(M)})} \right) 1 \left[N \leq \bar{q}_i^{-\frac{1}{2}} \right] \right] &\geq (1 - \epsilon) \mathbb{E} \left[(N_1 q_i^{(1)} + \dots + N_M q_i^{(M)}) 1 \left[N \leq \bar{q}_i^{-\frac{1}{2}} \right] \right] \\ &\geq (1 - \epsilon) q_i \mathbb{E}[\mathcal{T}_2 - \mathcal{T}_1] - (1 - \epsilon) \bar{q}_i \mathbb{E} \left[N 1 \left[N > \bar{q}_i^{-\frac{1}{2}} \right] \right]. \end{aligned}$$

Then, since $\mathbb{E}N < \infty$ and $1/\sqrt{\bar{q}_i} \rightarrow \infty$ as $i \rightarrow \infty$, it is straightforward to conclude that $\mathbb{E}[N 1[N > 1/\sqrt{\bar{q}_i}]] = \mathbb{E}N - \mathbb{E}[N 1[N \leq 1/\sqrt{\bar{q}_i}]] \rightarrow 0$ as $i \rightarrow \infty$, and, therefore, in conjunction with (22), we obtain

$$\mathbb{E} \left[1 - e^{-(N_1 q_i^{(1)} + \dots + N_M q_i^{(M)})} \right] \gtrsim (1 - \epsilon) q_i \mathbb{E}[\mathcal{T}_2 - \mathcal{T}_1],$$

as $i \rightarrow \infty$. Finally, after letting $\epsilon \rightarrow 0$ in the previous expression and (21), we complete the proof of this lemma. \diamond

Acknowledgements

We thank an anonymous reviewer for his/her helpful comments.

References

- [1] M. Abrams and R. Wooster. Proxy caching that estimates edge load delays. In *Proceedings of 6th International World Wide Web Conference*, Santa Clara, CA, April 1997.
- [2] V. Almeida, A. Bestavros, M. Crovella, and A. de Oliviera. Characterizing reference locality in the WWW. In *Proceedings of the Fourth International Conference on Parallel and Distributed Information Systems*, Miami Beach, Florida, December 1996.
- [3] M. Arlitt and C. Williamson. Web server workload characteristics: The search for invariants. In *Proceedings of ACM SIGMETRICS'1996*, Philadelphia, PA, May 1996.
- [4] F. Baccelli and P. Brémaud. *Elements of Queueing Theory*. Springer-Verlag, 2002.
- [5] O. Bahat and A. M. Makowski. Optimal replacement policies for non-uniform cache objects with optional eviction. In *Proceedings of IEEE INFOCOM'2003*, San Francisco, California, USA, April 2003.
- [6] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and Zipf-like distributions: Evidence and implications. In *Proceedings of IEEE INFOCOM'1999*, New York, NY, March 1999.
- [7] P. Cao and S. Irani. Cost-aware WWW proxy caching algorithms. In *Proceedings of the USENIX'1997 Annual Technical Conference*, Anaheim, California, January 1997.
- [8] Brian D. Davison. Web Caching and Content Delivery Resources. In <http://www.web-caching.com>.

- [9] P. R. Jelenković and A. Radovanović. Asymptotic Insensitivity of Least-Recently-Used Caching to Statistical Dependency. In *Proceedings of IEEE INFOCOM'2003*, San Francisco, April 2003.
- [10] P. R. Jelenković and A. Radovanović. Least-Recently-Used Caching with Dependent Requests. *Theoretical Computer Science*, 326(1-3):293–327, 2004.
- [11] P. R. Jelenković, X. Kang, and A. Radovanović. Near optimality of the discrete persistent access caching algorithm. *Discrete Mathematics and Theoretical Computer Science*, AD:201–222, 2005.
- [12] P. R. Jelenković and A. Radovanović. The Persistent-Access-Caching Algorithm. *Random Structures and Algorithms*, 33(2):219–251, May 2008.
- [13] S. Jin and A. Bestavros. GreedyDual* Web Caching Algorithm. In *Proceedings of the 5th International Web Caching and Content Delivery Workshop*, Lisbon, Portugal, May 2000.
- [14] S. Jin and A. Bestavros. Sources and characteristics of Web temporal locality. In *Proceedings of Mascots'2000: The IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, San Francisco, CA, August 2000.
- [15] E. G. Coffman Jr. and P. J. Denning. *Operating Systems Theory*. Prentice-Hall, 1973.
- [16] B. Moret and H. Shapiro. *Algorithms from P to NP: Volume 1 Design and Efficiency*. The Benjamin/Cummings Publishing Company, Redwood City, CA, 1991.