Exact Analysis of TTL Cache Networks

Daniel S. Berger[a], Philipp Gland[b], Sahil Singla[c], Florin Ciucu[d]

[a] *Distributed Computer Systems Lab, University of Kaiserslautern*
[b] *Institute for Mathematics, Technical University Berlin*
[c] *School of Computer Science, Carnegie Mellon University*
[d] *Computer Science Department, University of Warwick*

Abstract

TTL caching models have recently regained significant research interest due to their connection to popular caching policies such as LRU. This paper advances the state-of-the-art analysis of TTL-based cache networks by developing two *exact* methods with orthogonal generality and computational complexity. The first method generalizes existing results for line networks under renewal requests to the broad class of caching policies whereby evictions are driven by stopping times; in addition to classical policies used in DNS and web caching, our *stopping time model* captures an emerging new policy implemented in SDN switches and Amazon web services. The second method further generalizes these results to feedforward networks with Markov arrival process (MAP) requests. MAPs are particularly suitable for non-line networks because they are closed not only under superposition and splitting, as known, but also under caching operations with phase-type (PH) TTL distributions. The crucial benefit of the two closure properties is that they jointly enable the first exact analysis of TTL feedforward cache networks in great generality. Moreover, numerical results highlight that existing Poisson approximations in binary-tree topologies are subject to relative errors as large as 30%, depending on the tree depth.

Keywords: Cache Networks, TTL Caches, Markov Arrival Process

1. Introduction

Time-to-Live (TTL) caches decouple the eviction mechanisms amongst objects by associating each object with a timer. When a timer expires, the corresponding object is evicted from the cache. This seemingly simple scheme explicitly guarantees (weak) consistency, for which reason it has been widely deployed in DNS and web caching. What has recently however made TTL analytical models quite popular is a subtle mapping between *capacity-driven* (e.g., Least-Recently-Used (LRU)) and *TTL-based* caching policies. This mapping was firstly established through a remarkably accurate approximation by Che *et al.* [14] for the popular LRU policy, which was recently theoretically justified and extended to FIFO (first-in-first-out) and RND (random eviction) policies (Fricker *et al.* [24]), and further confirmed to hold for broader arrival models (Bianchi *et al.* [8]), and even in networks with several replication strategies (Martina *et al.* [37]). Moreover, Fofack *et al.* [15] independently argued that TTL caches capture the properties of LRU, FIFO, and RND policies, and, remarkably, presented the first exact analysis for a line of TTL caches. While the analysis of TTL caches is arguably simpler and more general than the analysis of capacity-driven policies, the exact analysis of TTL networks in general has remained an open problem.

When considering a cache network (including TTL-based), there are two inherent network operations which complicate the analysis: *input-output* and *superposition*. Given a caching node serving a request (point) process for some object (i.e., the *input*), the corresponding miss process is a sample of the request process at those points when the object is absent from the cache. The exact characterization of the *output* process is challenging: for instance, the convenient and often assumed memorylessness property of request processes is generally not retained by the corresponding miss process due to the TTL's inherent filtering effect. The *superposition* operation occurs when merging miss processes from upstream caches into a new input process. Since convenient statistical properties of the input processes (e.g., the renewal property) are altered through superposition, the analytical tractability of input-output operations in cache networks subject to superposition (e.g., trees) is conceivably more complex than in the case of line networks.

In this paper we provide the first exact analysis of caching (feedforward) networks by jointly addressing broad classes of request models, TTL distributions, and caching policies. The request processes are either renewals or Markov arrival processes (MAPs); the latter are dense in a suitable class of point processes and generalize, in particular, the more popular Markov-modulated Poisson processes. The TTLs follow general distributions including phase-type (PH), which are dense within the set of probability distributions on $[0, \infty)$. Moreover, we consider an abstract model for TTL caching policies, whereby cache evictions are driven by stopping times, and which captures in particular three popular policies. The '$\mathcal{R}$' policy regenerates the TTLs at every object's request and maps to the LRU policy. The 'Σ' policy regenerates the TTLs only at those requests resulting in cache misses and maps to FIFO and RND policies. Besides '$\mathcal{R}$' and 'Σ', which have already been studied, our stopping time model covers an emerging new policy, called '$\min(\Sigma, \mathcal{R})$', which combines the key features of Σ and $\mathcal{R}$, i.e., weak consistency guarantee and efficient utilization of cache space (i.e., both high hit ratio and low cache occupancy), respectively; this policy has been recently implemented in both SDN switches and Amazon web services.

We structure our results in two parts. First we generalize the recent results from Fofack *et al.* [15, 16] which cover line networks, renewals requests, general TTL distributions, and the '$\mathcal{R}$' and 'Σ' policies, by additionally covering the emerging '$\min(\Sigma, \mathcal{R})$' policy. The key analytical contribution is a unified method to recursively characterize input-output operations. This method is based on a suitable change of measure technique using martingales to derive the Laplace transform of a stopped sum, whereby the sum's stopping time characterizes the caching policy. Leveraging certain martingale properties to extend from deterministic to stopping times, we are able to systematically analyze the three caching policies and conceivably many others[1]. The proposed method suffers however from the same limitation as [15, 16]: since renewals are not closed under superposition, unless Poisson, only lines of caches can be (exactly) analyzed.

To address the annoying limitation of renewals' lack of superposition closure, the second part of the paper advocates MAPs to model request processes. The motivation to use MAPs is fairly straightforward since MAPs are known to be closed under superposition. What is remarkable, however, is that we are able to prove that MAPs are also closed under the input-output operation when the TTLs are described by PH distributions, for all the three '$\mathcal{R}$', 'Σ', and '$\min(\Sigma, \mathcal{R})$' policies. In other words, miss processes are also MAPs and trees of cache nodes can be iteratively analyzed. As a side remark, MAPs are also closed under a splitting operation, enabling thus the analysis of feedforward networks.

The two proposed methods advance the state-of-the-art analysis of TTL cache networks by providing the first exact results covering broad request models, caching policies, and network topologies. The second method in particular has the key feature of enabling the first (exact) analysis of feedforward networks and is thus conceivably more general than the first one. However, there is a fundamental tradeoff between the two, which is driven by the state explosion of MAPs under the superposition and input-output operations. Therefore, while the first method is computationally fast but restricted to line networks, the second has a much wider applicability but suffers from a high computational complexity, i.e., exponential in the number of caches. However, for our numerical results, we employ appropriate numerical methods which leverage the sparse nature of MAPs' matrices.

The rest of the paper is organized as follows. In Section 2 we summarize the cache models and discuss related work. In Section 3 we list model definitions and key objectives for the analysis. In Section 4 we present the change of measure technique to address lines of caches with renewal requests. In Section 5 we consider more general networks with MAP requests. We provide numerical results in Section 6 and conclusions in Section 7. All proofs and further examples of MAP input-output constructions are given in the Appendix.

[1] This apparently heavy technical machinery is employed to solve the difficult problem of deriving the Laplace transform of a *stopped* random walk (i.e., $X_1 + X_2 + \cdots + X_\tau$, where τ is a stopping time depending on X_i's); currently, higher moments for such stopped sums are only known in terms of bounds (see Gut [28], p. 22).

2. Cache Models and Related Work

Caching is implemented in many computer and communication systems, such as CPUs, databases, or content distribution networks. Consequently, the performance of caching systems has been extensively studied through many analytical models and techniques, some of which being discussed next.

Caching policies can roughly be divided into two groups: *capacity-driven* and *TTL-based* (see Rizzo and Vicisano [42]). In the former, objects' evictions are driven by the arrivals of uncached objects and the capacity constraint. In the latter, objects' evictions are determined by individual timers. When compared, TTL-based cache models are typically easier to analyze because the caching behavior of different objects is decoupled and can be thus represented in terms of independent point processes.

In this paper we address the following three TTL-based caching policies, which differ in the behavior of the TTLs' resets and eviction times:

1. *Policy* $\mathcal{R}$: The TTL is reset with every request and an object is evicted upon the TTL's expiration.
2. *Policy* Σ: The TTL is reset only at the times of unsuccessful requests and an object is evicted upon the TTL's expiration.
3. *Policy* $\min(\Sigma, \mathcal{R})$: Two TTLs are reset in parallel according to the $\mathcal{R}$ and Σ policies, respectively, and an object is evicted upon the expiration of *either* of them.

The $\mathcal{R}$ policy is particularly efficient in managing cache space: unpopular objects are quickly evicted from the cache, whereas popular object can dwell for much longer due to frequent TLL renewals. While $\mathcal{R}$ yields a high hit ratio, it suffers however from the lack of consistency guarantees, especially in the case of popular objects. In relationship to capacity-driven policies, $\mathcal{R}$ is the TTL-based correspondent of LRU caches (Che *et al.* [14] and Fricker *et al.* [24]). The general renewal formulation (both arrivals and TTLs are random variables) is due to Fofack *et al.* [15].

Unlike $\mathcal{R}$, the Σ model achieves weak consistency guarantees for objects undergoing updates at their origin [7], at the expense of a lower hit ratio. That is because TTLs bound the lifetime of outdated objects, but also unnecessarily preserve copies of unpopular objects. Σ corresponds to FIFO and RND (see, e.g., Fricker *et al.* [24]) and has recently been studied in the context of DNS by Fofack *et al.* [16]. As expected, the differences between $\mathcal{R}$ and Σ carry over to their capacity-driven correspondents (see Martina *et al.* [37]).

Finally, the $\min(\Sigma, \mathcal{R})$ policy enables a tradeoff between the weak consistency guarantee of Σ and the hit ratio efficiency of $\mathcal{R}$. The lifetime of possibly outdated popular objects is upper bounded by the Σ TTL; in turn, unpopular objects are quickly removed when the $\mathcal{R}$ TTL expires. Unlike $\mathcal{R}$ and Σ, the $\min(\Sigma, \mathcal{R})$ policy has not yet been formalized, although related implementations exist. A recent example is the flow eviction mechanism in the flow table of software defined networking (SDN) switches. In the popular OpenFlow switch specification [2], the "switch flow expiry mechanism" is defined by associating an "idle_timeout" ($\mathcal{R}$ TTL) and a "hard_timeout" (Σ TTL) with each flow in the table. Other examples can be found in Amazon ElastiCache's mechanism to achieve a good memory tradeoff [3], and the Squid web cache which uses a local "Storage LRU Expiration Age" ($\mathcal{R}$ TTL) and a Σ TTL that is set by content owners [1][2].

Early analytical models addressed capacity-driven single caches, which, contrary to their implementation simplicity, proved to be difficult to analyze. Some of the classic works (e.g., King [33] and Gelenbe [27]) provided exact results for LRU, FIFO, and RND policies. However, these results were argued to be intractable by Fagin and Price [23], Dan and Towsley [21], and Jelenkovic [30], who proposed instead accurate and computationally fast approximations.

With respect to the second group, namely TTL-based caches, the first analytical model for a single Σ cache under renewal arrivals and deterministic TTLs was given by Jung *et al.* [32] who derived the steady-state hit probability. Under the same assumptions, Bahat and Makowski [7] extended this result to the case of non-zero delays between the origin server and the cache, and derived the hit probability for those requests which are consistent with documents undergoing updates. For single $\mathcal{R}$ and Σ caches (for the latter

[2] There are conceivably many compound caching policies that can be described by our general stopping time model; in addition to the more 'classical' $\mathcal{R}$ and Σ, the study of the $\min(\Sigma, \mathcal{R})$ model is motivated by its recent applicability.

in parallel to our work), Fofack *et al.* [15, 16] obtained the exact hit probability and other caching metrics for renewal arrivals.

Concerning the analysis of cache networks, Rosenzweig *et al.* [44] applied the approximation scheme from [21] to networks of LRU caches (at the expense however of errors in the number of misses of up to 16% in certain torus networks). It is worth pointing out that the authors also proposed a Markov chain methodology, enabling the derivation of the inter-miss times at an LRU cache in terms of a PH distribution, and which can be regarded as a precursor of our general MAP methodology. In another recent work, Psaras *et al.* [41] proposed a Markov chain approximation for LRU caches which can then be linked together to form tree networks by assuming each cache's miss process to be Poisson. Under a similar approximation scheme (i.e., each cache's request process is Poisson) Gallo *et al.* [26] considered homogeneous tree networks under the RND policy.

A connection between the two domains of capacity-driven and TTL-based policies was established by the *characteristic time model* first introduced by Che *et al.* [14] for a simple two-level LRU cache network. This characteristic time model was recently refined by Fofack *et al.* [17] by ingeniously leveraging Little's law to relate the cache occupancy to the hit ratio. With the strong case made by Fricker *et al.* [25, 24] on its wide applicability, this mapping has recently gained significant popularity. In particular, its impressive accuracy and generality was confirmed by Bianchi *et al.* [8], Martina *et al.* [37], and Roberts and Sbihi [43], and parallel extensions to several other caching policies and replication strategies have also been proposed.

The success of the characteristic time model is due to a subtle mapping from the domain of capacity-driven caches to the domain of TTL-based caches. In the case of LRU, the key idea is to couple the cache capacity with the durations that objects spent in a cache under the condition that no further arrivals occur (i.e., $\mathcal{R}$). By assuming these (random) durations (called the characteristic time) as deterministic and equal for every object, the LRU model reduces to a TTL model [14, 24] that is easier to analyze.

To analyze TTL cache networks (e.g., as arising from the characteristic time model for a capacity-driven policy), Martina *et al.* [37] rely on Poisson approximations of the output processes (for both $\mathcal{R}$ and Σ caches) and report accurate results. Moreover, Fofack *et al.* [15, 16] derived the first exact results for a line of $\mathcal{R}$ and Σ caches under renewal requests and analyzed tree networks by relying on a seemingly accurate renewal approximation of the superposed processes; more general topologies have been recently addressed in Fofack *et al.* [17] using a renewal approximation of superposed processes based on moments matching.

Unlike these works, which assume an independent cache behavior, another set of works considered hierarchies of caches where the layers are synchronized by an aging mechanism: the TTL values at child caches are set to coincide with the remaining TTLs of parent caches. In this way, Cohen and Kaplan [19] were able to derive the miss rate for a two-level hierarchy, and Cohen *et al.* [18] extended this result to heterogeneous parent nodes. Remarkably, by ingeniously leveraging the system's property that misses occur synchronously, Hou *et al.* [29] were able to analyze trees of caches under Poisson arrivals at the leaf caches.

3. Roadmap

In this section we state the key objectives for analyzing lines of caches and feedforward networks. First we give some general definitions concerning some arbitrary node in a cache network.

Definition 1 (Arrival/Input Process).
For each object, the arrival process is represented as a point (counting) process $N(t)$. The corresponding inter-arrival process is denoted by $\{X_t\}_{t\geq 1}$.

When analyzing a cache network, one needs to characterize the miss/output process relating two consecutive caches. We give the definition in the renewal case.

Definition 2 (Miss/Output Process).
Let the inter-request times and TTLs to some caching node be given by the two independent renewal processes $\{X_t\}_{t\geq 1}$ and $\{T_t\}_{t\geq 1}$. The corresponding miss process is also a renewal process with the same distribution as the stopped sum

$$S_\tau := X_1 + \cdots + X_\tau \ , \tag{1}$$

where τ is a stopping time defined separately for each caching (TTL) policy:

$$\begin{aligned} \textit{Policy } \mathcal{R}: \quad & \tau & := \min\{t : X_t > T_t\} \qquad (2) \\ \textit{Policy } \Sigma: \quad & \tau & := \min\{t : \sum_{s=1}^{t} X_s > T_1\} \qquad (3) \\ \textit{Policy } \min(\Sigma, \mathcal{R}): \quad & \tau & := \min\{\min\{t : \sum_{s=1}^{t} X_s > T_1^{\Sigma}\}, \min\{t : X_t > T_t^{\mathcal{R}}\}\} \ . \qquad (4) \end{aligned}$$

For the last policy, T_1^{Σ} and $T_t^{\mathcal{R}}$ are independent renewal processes.

The corresponding definition for request processes other than renewal, such as MAPs, can be stated similarly and is omitted for brevity.

As a side remark, the structure of the first two stopping times justifies the notation for the caching policies by $\mathcal{R}$ and Σ: the former has a renewal structure (whence the letter $\mathcal{R}$), whereas the latter has a sum structure (whence the letter Σ). We incorporate this notation in the whole notation of a caching node, by borrowing from Kendall's notation in queueing theory.

Notation 1 (Caching Node)**.**
Depending on the TTL policy, a caching node is denoted as either one of the following triplets G-G-$\mathcal{R}$, G-G-Σ, or G-G-min$(\mathcal{R}, \Sigma)$, where the two G's stand for the generic distributions of the inter-arrival times and the TTLs, respectively.

As an example, a cache with exponentially distributed inter-arrival and TTL times, and implementing the $\mathcal{R}$ policy, is denoted by *M-M-$\mathcal{R}$*. Some other distributions used in this paper are the deterministic (D) case, the exponential (M), and the phase-type (PH) distribution.

Cache performance is commonly measured in terms of hit/miss probabilities, which indicate the improvement in link utilization when using a cache.

Definition 3 (Hit/Miss Probability)**.**
Consider an arbitrary cache with arrival process $N(t)$ and miss process $M(t)$, for some fixed object. The hit and miss probabilities are defined as

$$H := \lim_{t\to\infty} \left(1 - \frac{M(t)}{N(t)}\right) \quad \textit{and} \quad M := \lim_{t\to\infty} \frac{M(t)}{N(t)} \ ,$$

respectively, subject to convergence.

Another relevant metric is the cache occupancy which defines the average amount of storage required by an object, and also establishes a connection between capacity-driven and TTL-based cache models through a suitable set of parameters. For example, the connection between an LRU cache's capacity and the corresponding TTL-$\mathcal{R}$ model is established by equalizing the LRU cache's capacity with the summation of the occupancies $\pi(o)$ of the objects o in the TTL cache (as used in [14, 24, 8, 37, 43])

$$C = \sum_{o\in P} \pi(o) = \sum_{o\in P} \mathbb{E}\left[1_{\{o \text{ in cache}\}}\right] \ , \qquad (5)$$

where P represents the objects' population.

Definition 4 (Cache Occupancy)**.**
Let $C(t)$ be a random binary process representing whether the object is in the cache or not at time t. The cache occupancy, omitting the object's index, is defined as

$$\pi := \lim_{t\to\infty} \frac{\int_0^t C(s)ds}{t} \ ,$$

subject to convergence.

We next briefly introduce the key objectives for analyzing lines and feedforward networks.

3.1. Lines of Caches with Renewal Arrivals

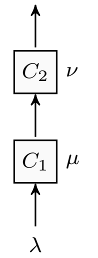


Figure 1: A line of two caches

Consider the simplified line network with two nodes from Figure 1. At the first node, requests for some object arrive according to a renewal process $\{X_t\}_{t\geq 1}$. If the object is in the cache at the time of a request, then the request is successful and the object is fetched. Otherwise, for every unsuccessful request at some node, the object is recursively requested at the next node in the line. Once the object is successfully found at some downstream node, it is (instantaneously) transferred to all upstream nodes. For the model's completeness we assume that the last node always has a copy of the object. Moreover, all the nodes implement either $\mathcal{R}$, Σ, or $\min(\Sigma, \mathcal{R})$, and for the sake of generality different nodes can implement different policies.

We address a single node with a given arrival/input process and TTL distribution. For this setting our objective is to derive the Laplace transform of the corresponding miss/output process, i.e., of the stopped random walk S_τ from Eq. (1):

$$\mathcal{L}(S_\tau) := E\left[e^{-\omega(X_1+X_2+\cdots+X_\tau)}\right] , \tag{6}$$

for some $\omega > 0$.

This technique can be iteratively applied along an entire line of caches using numerical methods (as in Fofack *et al.* [15, 16]); numerical methods are not necessary in the case of exponential TTLs. We emphasize that, unlike [15, 16] which are restricted to $\mathcal{R}$ and Σ, our method additionally covers the emerging $\min(\Sigma, \mathcal{R})$.

3.2. Feedforward Networks with MAP Arrivals

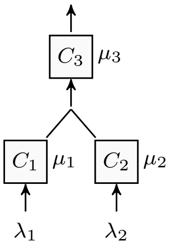


Figure 2: A tree of caches

The previous technique suffers from the major limitation that cache requests must be a renewal process and thus it does not apply to more general topologies subject to a superposition operation. Indeed, consider the simple tree topology from Figure 2 in which the inter-request times at the leaf caches are $\exp(\lambda_1)$ (exponential) and $\exp(\lambda_2)$, and the corresponding TTLs are $\exp(\mu_1)$ and $\exp(\mu_2)$, respectively; all the processes are statistically independent. The inter-miss times at the leaf caches are renewal processes with $hypo(\lambda_1, \mu_1)$ (hypoexponential) and $hypo(\lambda_2, \mu_2)$ distributions. The superposition of the two renewal processes is not a renewal process, which means that the technique targeting lines of caches does not apply at the root cache (the superposition of renewal processes is a renewal process if and only if the superposed processes are Poisson).

Let us now recall the two main analytical operations which must be accounted for when analyzing trees of caches:

1. *input-output*: the characterization of the inter-miss process from the inter-request process.
2. *superposition*: the characterization of a single inter-request process from multiple ones (e.g., at the root cache from Figure 2).

Unlike the *input-output* operation which will be shown in Section 4 to be tractable (yet subject to recursions and also the evaluation of convolutions in the case of Σ and $\min(\Sigma, \mathcal{R})$ caches), the *superposition* operation is conceivably the bottleneck due to the lack of closure of renewal processes. To circumvent this apparent difficulty, the natural generalization of renewal processes are Markov arrival process (MAPs), which are known to be closed under *superposition* (and also splitting). The remaining objective is to additionally show that MAPs are also closed under the *input-output* operation of caches (see Section 5).

We point out that the idea of using MAPs has been efficiently used in the past to model systems with non-renewal behavior, e.g., single queues with non-renewal arrivals (Lucantoni *et al.* [36]) or closed queueing networks with non-renewal workloads (Casale *et al.* [13]); for an excellent related survey see Asmussen [4].

4. Lines of Caches

In this section we propose a unified method to analyze lines of G-G-$\mathcal{R}$, G-G-Σ, and G-G-$\min(\Sigma,\mathcal{R})$ caches. First we instantiate the caching metrics from Definitions 3 and 4 for the renewal case.

Lemma 1 (Hit/Miss Probabilities (proof in Appendix 8.1))**.**
For G-G-$\mathcal{R}$, G-G-Σ, and G-G-min$(\mathcal{R},\Sigma)$ caches, the hit and miss probabilities from Definition 3 are

$$H = \frac{\mathbb{E}[\tau]-1}{\mathbb{E}[\tau]} \quad \textit{and} \quad M = \frac{1}{\mathbb{E}[\tau]} \ . \tag{7}$$

In particular, for G-G-$\mathcal{R}$, it holds $H = \mathbb{P}(X \leq T)$ and $M = \mathbb{P}(X > T)$.

The expression of the miss probability for the G-G-Σ cache is the same as in Jung *et al.* [32]. Unlike in the G-G-$\mathcal{R}$ case, $E[\tau]$ for the other two policies cannot be generally given in closed-form due to the underlying convolution in the definition of τ from Eqs. (3)-(4); it is, however, often straightforward to derive $E[\tau]$ for particular distributions of $\{X_t\}_{t\geq 1}$ and $\{T_t\}_{t\geq 1}$.

Lemma 2 (Cache Occupancy (proof in Appendix 8.2))**.**
For the G-G-$\mathcal{R}$, G-G-Σ, and G-G-$\min(\Sigma,\mathcal{R})$ caches, the cache occupancies from Definition 4 are

$$\pi_{\mathcal{R}} = \frac{\mathbb{E}[min\{X,T\}]}{\mathbb{E}[X]}, \quad \pi_{\Sigma} = \frac{\mathbb{E}[T]}{\mathbb{E}[S_\tau]}, \quad \pi_{\min(\Sigma,\mathcal{R})} = \frac{E\left[\min\{\sum_{s=1}^{\tau}\min\{X_s,T_s^{\mathcal{R}}\},T_1^{\Sigma}\}\right]}{E[S_\tau]} \ .$$

The expression for G-G-$\mathcal{R}$ is the same as the one given by Fofack *et al.* [15] (written therein in the equivalent form $\pi_{\mathcal{R}} = \mathbb{E}\left[\int_0^X \mathbb{P}(T > t)dt\right]/\mathbb{E}[X]$). Note that the last expectation depends on the stopping time τ from Eq. (4), whose mass function is later provided in Corollary 3; moreover, to compute the cache occupancy, a decoupling argument like the one we provide for the transforms of the inter-miss times is needed to avoid the implicit correlations amongst the stopping time, the inter-arrival times, and the TTLs (for all $\mathcal{R}$, Σ, and $\min(\Sigma,\mathcal{R})$).

The key problem to compute caching metrics at the downstream nodes in a line of caches is to recursively characterize the inter-miss time S_τ defined in Eq. (1). The expression of S_τ, as well as those of the above caching metrics, suggests following a martingale based technique to characterize S_τ. This (rough) idea is driven by the fact that stopping times—which are at the core of the very definition of S_τ—preserve certain martingale results, e.g., if L_t is a martingale and τ is a bounded stopping time then $E[L_\tau] = E[L_1]$, which is a particular case of the optional stopping theorem. Note however that caching stopping times are generally not necessarily bounded, and thus a technical probability framework is needed. For instance, the stopping times from Eqs. (2)-(4) are under realistic assumptions almost surely finite but may be unbounded.

Next we demonstrate the effectiveness of relying on martingale techniques to derive an elegant and unified analysis of G-G-$\mathcal{R}$, G-G-Σ, and G-G-$\min(\Sigma,\mathcal{R})$ caches. To this end, we first provide a closed-form result (in most of the cases) for the Laplace transform $\mathcal{L}(S_\tau)$ of the stopped random sum S_τ. This result will be instrumental to the analysis of the $\mathcal{R}$, Σ, and $\min(\Sigma,\mathcal{R})$ policies.

4.1. The Laplace Transform $\mathcal{L}(S_\tau)$ of a Stopped Sum

Consider the two independent renewal processes $\{X_t\}_{t\geq 1}$ and $\{T_t\}_{t\geq 1}$ on a joint probability space $(\Omega,\mathcal{F},\mathbb{P})$ (e.g., as in Definition 2). Denote the corresponding distribution functions by $F(x)$ and $G(x)$, and assume the existence of corresponding densities $f(x)$ and $g(x)$, respectively. Let $\mathcal{F}_t = \sigma((X_1,T_1),\ldots,(X_t,T_t))$, $\mathbb{F} = (\mathcal{F},\{\mathcal{F}_t\}_{t\geq 1})$ denote the filtration associated with S_τ, and let $(\Omega,\mathbb{F},\mathbb{P})$ denote the corresponding filtered probability space. When clear from the context the time indexes are suppressed.

Next we provide a closed-form expression for the Laplace transform $\mathcal{L}(S_\tau)$ of the stopped sum S_τ from Eq. (6). Recall from Eqs. (2)-(4) that τ is a stopping time with respect to the filtration $\mathcal{F}_t$. One may remark that, due to the intrinsic dependencies amongst X_t's and the stopping time τ, the analysis of the stopped sum S_τ is conceivably quite involved even for stopping times w.r.t. the filtration $\mathcal{F}'_t = \sigma(X_1,X_2,\ldots,X_t)$. In fact, unlike the first moment which is relatively easily obtained as Wald's equation, i.e., $E[S_\tau] = E[\tau]E[X_1]$ (under the additional condition that $E[\tau] < \infty$), higher moments, however, are typically only known in terms of bounds (see Gut [28], p. 22).

Despite the apparent technical difficulties, we will next show that $\mathcal{L}(S_\tau)$, for stopping times τ w.r.t. $\mathcal{F}_t$, can be derived in a rather straightforward manner. The key idea is to construct a suitable new filtered probability space $(\Omega, \mathbb{F}, \tilde{\mathbb{P}})$, whereby the new probability measure $\tilde{\mathbb{P}}$ decouples the dependencies amongst X_t's and τ. Informally, the key idea to compute $\mathcal{L}(S_\tau)$ in closed-form is to *offshore* the underlying derivations to the new $(\Omega, \mathbb{F}, \tilde{\mathbb{P}})$ space.

This technique is known as *change of measure*. The change of measure itself (e.g., from $\mathbb{P}$ to $\tilde{\mathbb{P}}$) is performed as such *measures* in the original space (e.g., $\mathbb{P}(A)$ for $A \in \mathcal{F}$) can be obtained in terms of the new (changed) measure in a much simpler manner. An example of an application of this technique is in rare events simulations, whereby rare events become more likely to occur under the new (changed) measure, or more precisely under the new (changed) density, guaranteeing thus faster convergence speeds than Monte-Carlo simulations (see Pham [40]). Another application is in pricing risks in incomplete markets, by constructing a new risk-neutral probability measure (see Cox *et al.* [20]). Such risk-neutral measures have also been constructed in financial models, in order to simplify a model with drift into one with constant expectation and allowing thus the application of the Girsanov theorem to describe the process dynamics (see Musiela and Rutkowski [38]). Another application is an elegant proof for Cramér's theorem in large deviation theory (see Dembo and Zeitouni [22], p. 27).

To perform the intended change of measure, we extend the measure construction for a filtration $\mathcal{F}'_t = \sigma(X_1, X_2, \ldots, X_t)$ (see Asmussen [5], p. 358) to the product filtration $\mathcal{F}_t = \sigma((X_1, T_1), \ldots, (X_t, T_t))$. While the extension proceeds mutatis mutandis, mainly due to the independence between $(X_t)_{t\geq 1}$ and $(T_t)_{t\geq 1}$, the key to our construction is to only *tilt* the distribution $F(x)$ of X_t while preserving the distribution $G(x)$ of T_t; for this reason, we refer to our change of measure as a *fractional* change of measure.

Definition 5 (Fractional Change of Measure)**.**
For any $t \geq 1$ and $F \in \mathcal{F}_t$ define the tilted probability measure $\tilde{\mathbb{P}}_t$ as

$$\tilde{\mathbb{P}}_t(F) := E\left[L_t 1_F\right] \ , \tag{8}$$

where L_t is the Wald's martingale w.r.t. the filtered space $(\Omega, (\mathcal{F}, \{\mathcal{F}'_t\}_{t\geq 1}), \mathbb{P})$, defined for some $\omega > 0$ as

$$L_t := \frac{e^{-\omega S_t}}{\mathcal{L}(X)^t} \ . \tag{9}$$

The tilted measures $\tilde{\mathbb{P}}_t$, which are by construction restricted to $\mathcal{F}_t$, uniquely extend to a probability measure $\tilde{\mathbb{P}}$ on $\mathcal{F}$ which is Kolmogorov consistent, i.e., $\tilde{\mathbb{P}}(F) = \tilde{\mathbb{P}}_t(F) = E\left[L_t 1_F\right]$ for all $F \in \mathcal{F}_t$. The proof follows the proof of Proposition 3.1 from [5], with the observation that L_t is also a martingale w.r.t. the product filtered space $(\Omega, \mathbb{F}, \mathbb{P})$ due to the independence between $(X_t)_{t\geq 1}$ and $(T_t)_{t\geq 1}$.

Besides enabling the construction of the consistent probability measure $\tilde{\mathbb{P}}$ (mainly using the fact that L_t's are martingales with $E\left[L_t\right] = 1$), there are two technical reasons behind the fractional change of measure from Definition 5. On one hand, L_t corresponds to the Radon-Nikodym density of the Kolmogorov extended measure $\tilde{\mathbb{P}}$ (in addition to that of $\mathbb{P}_t$ as well) w.r.t. $\mathbb{P}$, i.e., $L_t = \frac{d\tilde{\mathbb{P}}}{d\mathbb{P}}$ on $(\Omega, \mathcal{F}_t)$ for all $t \geq 1$. This allows the computation of integrals w.r.t. $\tilde{\mathbb{P}}$ according to the integration rule

$$\int_A Y d\tilde{\mathbb{P}} = \int_A Y L_t d\mathbb{P} \ \forall A \in \mathcal{F}_t$$

for $\mathcal{F}_t$-measurable Y, under the condition that YL_t is integrable w.r.t. $\mathbb{P}$ (see Billingsley [9], Theorem 16.11). In particular, one has in terms of expectations

$$\tilde{E}[Y] = E[YL_t] \ , \tag{10}$$

where $\tilde{E}[\cdot]$ is the expectation w.r.t. $\tilde{\mathbb{P}}$.

On the other hand, the *particular* expression of L_t from Eq. (9) lends itself, by plugging in above $Y := \mathcal{L}(X)^t$ and cancelling out terms, to

$$E\left[e^{-\omega S_t}\right] = \tilde{E}\left[\mathcal{L}(X)^t\right] \ \forall t \geq 1 \ .$$

While this result is seemingly trivial, since the expectation $\tilde{E}$ can be dropped due to the non-randomness of $\mathcal{L}(X)$, it does capture the expression of the sought (final) result when t is replaced by a stopping time τ w.r.t. $\mathcal{F}_t$ (the proof follows similarly by applying Theorem 3.2 from Asmussen [5]).

Theorem 1 (Laplace Transform of S_τ (proof in Appendix 8.3))**.**
For an (a.s.) finite stopping time τ, the Laplace transform of the stopped sum S_τ from Eq. (1) is given by

$$\mathcal{L}(S_\tau) = \tilde{E}\left[\mathcal{L}(X)^\tau\right] \ . \tag{11}$$

Note that this result is a manifestation of the earlier stated motivation that 'stopping times preserve martingale properties', justifying thus our overall martingale framework to analyze the inter-miss times S_τ. The proof proceeds along the same lines as in [5] with the main difference of working on an extended product filtration. Theorem 1 herein is thus a simple extension of Theorem 3.2 from [5].

4.2. The Renewal Laplace Transform of Inter-Miss Times

Here we apply Theorem 1 to derive the particular transforms of the inter-miss times for our cache models. We remark that the results for the *G-G-*$\mathcal{R}$ and *G-G-*Σ caching models have previously been obtained separately in [15] and [16], respectively. Besides allowing to address practical composite cache policies (we consider *G-G-*min$(\mathcal{R}, \Sigma)$ as an example, here), our results allow the uniform analysis of *G-G-*$\mathcal{R}$ and *G-G-*Σ caching models.

In order to apply Theorem 1, note that the stopped sum S_τ from Definition 2 corresponds to the inter-miss time of a particular caching policy, which is given in terms of the stopping times τ from Eqs. (2)-(4). The crucial aspect is that the transform $\mathcal{L}(X)$ is computed w.r.t. the original probability measure $\mathbb{P}$. What remains to compute is τ's pmf under the changed measure $\tilde{\mathbb{P}}$. In other words, the computations for $\mathcal{L}(X)$ and the pmf of τ under $\tilde{\mathbb{P}}$ are entirely decoupled, circumventing thus the dependencies within S_τ.

To facilitate the auxiliary calculus under $\tilde{\mathbb{P}}$ we next give the following technical result whose proof is immediate from the definitions by using Fubini's theorem.

Proposition 1 (proof in Appendix 8.4)**.** *On the new probability space $(\Omega, \mathcal{F}, \tilde{\mathbb{P}})$, the random variables X_t and T_t have the following distribution functions for all $t \geq 1$ and $x \geq 0$*

$$\begin{aligned} \tilde{F}(x) &:= \tilde{\mathbb{P}}\left(X_t \leq x\right) = \frac{E\left[e^{-\omega X_t} 1_{\{X_t \leq x\}}\right]}{\mathcal{L}(X)} \\ \tilde{G}(x) &:= \tilde{\mathbb{P}}\left(T_t \leq x\right) = G(x) \ . \end{aligned}$$

The corresponding densities are $\tilde{f}(x) := d\tilde{F}(x) = \frac{e^{-\omega x} f(x)}{\mathcal{L}(X)}$ and $\tilde{g}(x) := d\tilde{G}(x) = g(x)$, respectively. Moreover, X_t and T_t remain independent under $\tilde{\mathbb{P}}$.

Using the integration rules from Proposition 1, it is now straightforward to compute the transform of the inter-miss time for the $\mathcal{R}$ model.

Corollary 1 (*G-G-*$\mathcal{R}$ (proof in Appendix 8.5))**.** [3]
Let τ as in Eq. (2). If $\psi(\omega) := \mathbb{E}\left[e^{-\omega X} 1_{\{X \leq T\}}\right] < 1$ for some $\omega > 0$, then the Laplace transform of the inter-miss time in the G-G-$\mathcal{R}$ model is given by

$$\mathcal{L}(S_\tau) = \frac{\mathcal{L}(X) - \psi(\omega)}{1 - \psi(\omega)} \ . \tag{12}$$

[3] This result was previously obtained by Fofack *et al.* [15].

To derive Corollary 1, it is sufficient to derive the probability mass function of τ under $\tilde{\mathbb{P}}$, which follows a simple geometric structure and is immediate from Proposition 1.

Corollary 2 (G-G-Σ (proof in Appendix 8.5))**.** [4]
Let τ as in Eq. (3). Then for some $\omega > 0$ the Laplace transform of the inter-miss time in the G-G-Σ model is given by

$$\mathcal{L}(S_\tau) = \sum_{t\geq 1} \phi(\omega)^t \mathbb{E}\left[\tilde{F}^{\,t-1}(T) \;-\; \tilde{F}^{\,t}(T)\right] \tag{13}$$

where $\tilde{F}^{\,t}$ is the distribution of the t-fold convolution of X in the tilted probability space $(\Omega, \mathcal{F}, \tilde{\mathbb{P}})$.

Unlike the G-G-$\mathcal{R}$ model, the G-G-Σ model is more tedious to analyze due to the expression of the stopping time τ from Eq. (3). In particular, to account for the sum in the expression of τ, a convolution density is required. The result for the G-G-min$(\mathcal{R}, \Sigma)$ cache model is notationally complex, and therefore is also stated in the Appendix, in Section 8.8.

These corollaries conclude the analysis of lines of renewal caches; numerical results will be provided in Section 6.

5. Feedforward Cache Networks

In this section we prove that MAPs are remarkably suitable to model inter-request processes in a feedforward cache network, to the point that the associated superposition and input-output operations are quite straightforward.

MAPs generalize Poisson processes by allowing the inter-arrival times to be dependent and also to belong to the broad class of phase-type (PH) distributions (to be defined later). MAPs have been motivated in particular by the need to mitigate the modelling restrictions imposed by the exponential distribution. From an analytical perspective, MAPs are quite attractive not only due to their versatility (MAPs are in fact dense in a large class of point processes, see Asmussen and Koole [6]), but also due to their tractability. Let us next give a common definition of MAPs.

Definition 6 (Markov Arrival Process (MAP))**.** *A Markov arrival process is defined as a pair of matrices $(\mathbf{D}_0, \mathbf{D}_1)$ with equal dimensions, or as a joint Markov process $(J(t), N(t))$. The matrix $Q := \mathbf{D}_0 + \mathbf{D}_1$ is the generator of a background Markov process $J(t)$. The matrix $\mathbf{D}_0$ is non-singular and a subintensity[5], and contains the rates of the so-called hidden transitions which govern the change of $J(t)$ only. In turn, the matrix $\mathbf{D}_1$ contains the (positive) rates of the so-called active transitions which govern the change of both $J(t)$ and a counting process $N(t)$, i.e., if $J(t^-) = i$ and a transition (i, j) from $\mathbf{D}_1$ occurs at time t, then $J(t) = j$ and $N(t) = N(t^-) + 1$.*

With abuse of notation we denote a MAP as $M = (\mathbf{D}_0, \mathbf{D}_1)$. For the sake of familiarizing with MAPs, let us represent a two-state Markov Modulated Poisson Process (MMPP), described in terms of a background Markov process $J(t)$ with two states (see Figure 3); depending on the state, arrivals can occur (and contribute to a counting process $N(t)$) at rates λ_1 and λ_2.

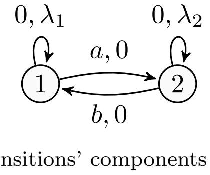


Figure 3: MAP representation of a MMPP; the transitions' components are hidden and active, respectively (e.g., in '$0, \lambda_1$', 0 is hidden and λ_1 is active), and competing with each other. Each active transition increments $N(t)$ by one unit.

[4] This result appeared in a parallel work by Fofack *et al.* [16]; in Appendix 8.7 we show the equivalence to our expression.

[5] A subintensity matrix S is similar to a stochastic matrix, except that rows sum to a non-positive value; formally, $S_{ii} < 0$, $S_{ij} \geq 0$ for $i \neq j$, and $\sum_{j=1}^{m} S_{ij} \leq 0$ $\forall i \in \{1, \ldots, m\}$.

The corresponding MAP is given by the hidden and active transition matrices

$$\mathbf{D}_0 = \begin{pmatrix} -a-\lambda_1 & a \\ b & -b-\lambda_2 \end{pmatrix},\ \mathbf{D}_1 = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}\ ,\ \text{where } \mathbf{D}_0 + \mathbf{D}_1 = \begin{pmatrix} -a & a \\ b & -b \end{pmatrix}$$

is the generator of $J(t)$.

For an excellent introduction to Markov Arrival Processes we refer to Casale [12].

5.1. *MAPs for Two Simple Cache Networks*

To introduce the main ideas of constructing MAPs for the input-output and superposition operations, we briefly present two simple examples of cache networks; the general results will be presented thereafter. For further more complex examples see Appendix 10.

5.1.1. *Input-Output*

We first illustrate the input-output operation in a line-network scenario as in Section 4. Let the network from Figure 1 consist of two Σ caching nodes C_1 and C_2; requests arrive at C_1 as a Poisson process with rate λ, and the TTLs are $exp(\mu)$ and $exp(\nu)$, respectively. At node C_1, the arrivals can be represented as a Poisson process $N(t)$, which is itself an elementary single-state MAP M_1 defined in terms of

$$\mathbf{D}_0 = (-\lambda),\ \mathbf{D}_1 = (\lambda)\ ,$$

and a background Markov process with generator $\mathbf{Q} = (0)$. See Figure 4.(a) for its graphical representation.

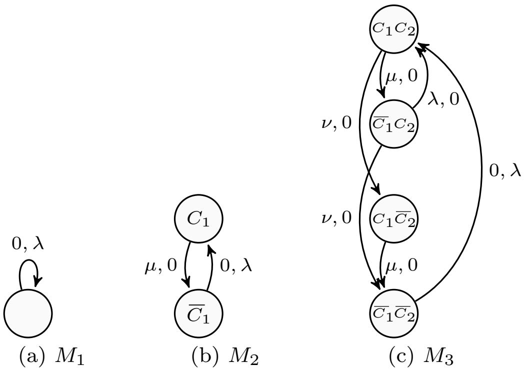


Figure 4: M_1 corresponds to the arriving Poisson MAP at cache C_1 in Figure 1, M_2 to the output of C_1, and M_3 to the output of C_2

To construct the arrival MAP M_2 at C_2, capturing the inter-miss times at C_1, the basic idea is to duplicate the states of M_1 and suitably construct the hidden and active transitions. The new states (see Figure 4.(b)) are denoted by $\overline{C}_1$ (with the interpretation 'object is not in the cache') and C_1 (with the interpretation 'object is in the cache'). While in state $\overline{C}_1$, an arrival to C_1 triggers a miss—whence the active transition λ (i.e., the second component of '$0, \lambda$') to C_1. While in state C_1, the TTL may expire and hence the hidden transition μ to $\overline{C}_1$. It is important to remark that external requests while in C_1 result in hits and thus do not affect the MAP. Note also that the constructed MAP recovers that the inter-miss times (i.e., the time between two active transitions) are $hypo(\lambda, \mu)$ (which is immediate from Theorem 1). In matrix form, M_2 can also be represented as

$$\mathbf{D}_0' = \begin{pmatrix} -\lambda & 0 \\ \mu & -\mu \end{pmatrix} \text{ and } \mathbf{D}_1' = \begin{pmatrix} 0 & \lambda \\ 0 & 0 \end{pmatrix}\ .$$

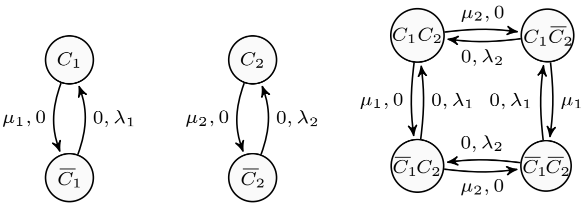
Applying the same idea, we construct M_3 by duplicating the states of M_2 (see Figure 4.(c)). The four new states have the interpretations 'object is in none of the caches' (state $\overline{C}_1\overline{C}_2$), 'object is in only one cache' (states $C_1\overline{C}_2$ and $\overline{C}_1C_2$), and 'object is in both caches' (state C_1C_2). There are two important observations to make: one is that there is a single active transition (i.e., '$0, \lambda$') from $\overline{C}_1\overline{C}_2$ to C_1C_2. The other is that while in $\overline{C}_1C_2$, a request at C_1 does not result in an active transition because the object is already in C_2—and no miss at C_2 can occur—whence the *hidden* transition '$\lambda, 0$'. The remaining transitions are all hidden, capturing all possible TTLs' expirations depending on the states. In matrix form, M'' can also be represented as

$$\mathbf{D}_0'' = \begin{pmatrix} -\lambda & 0 & 0 & 0 \\ \mu & -\mu & 0 & 0 \\ \nu & 0 & -\lambda-\nu & \lambda \\ 0 & \nu & \mu & -\mu-\nu \end{pmatrix} ,$$

and $\mathbf{D}_1''$ contains only zeros except for λ on position $(1, 4)$.

5.1.2. Superposition

Consider now the tree topology from Figure 2. Applying the previous ideas, we can immediately construct the (independent) MAPs M_1 and M_2 corresponding to the inter-miss times at the caches C_1 and C_2, respectively (see Figures 5.(a)-(b)).



(a) M_1 (b) M_2 (c) M_3

Figure 5: The MAPs M_1, M_2, and M_3 corresponding to the inter-miss times at caches C_1 and C_2 from Figure 2, and their superposition

The construction of the superposition of M_1 and M_2, denoted by M_3, proceeds by forming the Cartesian product of the sets of states of M_1 and M_2; the resulting states have the same interpretation as in the previous subsection, e.g., $C_1\overline{C}_2$ stands for 'object in cache C_1 and not in cache C_2'. Moreover, the formation of the hidden and active transitions proceeds as before. For instance, while in state $C_1\overline{C}_2$ two transitions are possible: an active one (i.e., '$0, \lambda_2$') corresponding to an arrival at C_2, and a hidden one (i.e., '$\mu_1, 0$') corresponding to the TTL expiration.

Furthermore, one can construct the MAP corresponding to the inter-miss times at cache C_3 (in Figure 2) following the ideas so far. As the resulting number of states is eight (i.e., from doubling the states of M_3), we omit the graphical depictions. We can remark however that both the input-output and superposition operations result in an exponential increase of the number of MAP states; this fact will be elaborated more precisely later.

5.2. General Results

We now present the general results for constructing MAPs in feedforward networks of $\mathcal{R}$, Σ, and $\min(\Sigma, \mathcal{R})$ caches. As we previously mentioned, the MAP framework allows TTLs to belong to the broad class of PH distributions, which we define next.

Definition 7 (Phase-Type Distribution). *Let* $\mathbf{S}$ *be a* $m \times m$ *subintensity,* $\mathbf{S}_0 := -\mathbf{S}\boldsymbol{1}$*, and* π *be a stochastic* m*-vector (note the abuse of notation for* π*). Define a Markov process with generator*

$$\mathbf{P} := \begin{pmatrix} 0 & \mathbf{0} \\ \mathbf{S}_0 & \mathbf{S} \end{pmatrix} ,$$

which extends $\mathbf{S}$ *by an absorbing state* 0 *and exit transitions from every state in* $\mathbf{S}$ *to* 0*. A PH distribution (of order* m*), denoted as* $T = (\mathbf{S}, \pi)$*, is defined as the time until absorption in state* 0 *of the Markov process generated by* P*, and which starts in any of the states* $\{1, \ldots, m\}$ *according to* π*.*

We remark that we chose the less standard notation with the **0** vector on the first row instead of the last; this choice will permit expressing the input-output cache operation in a convenient manner.

Next we summarize the known result of MAPs' superposition and then present our main results on the input-output cache operation involving MAP requests and PH TTLs.

5.2.1. Superposition

First we briefly review the superposition of MAPs, for which we need to introduce the Kronecker sum $\oplus$ and product $\otimes$ operators for matrices.

If $\mathbf{A}$ and $\mathbf{B}$ are $m \times m$ and $n \times n$ matrices then

$$\mathbf{A} \otimes \mathbf{B} := \begin{pmatrix} a_{11}\mathbf{B} & \cdots & a_{1m}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{n1}\mathbf{B} & \cdots & a_{mm}\mathbf{B} \end{pmatrix} \text{ and } \mathbf{A} \oplus \mathbf{B} := \mathbf{A} \otimes \mathbf{I}_n + \mathbf{I}_m \otimes \mathbf{B} \ ,$$

where $\mathbf{A} = (a_{i,j})$ and $\mathbf{I}_k$ is the $k \times k$ identity matrix (note: the operator $\oplus$ is simplified for the case of square matrices).

Theorem 2 (MAP Superposition (proof due to [36]))**.** *If the MAPs $M_1, \ldots, M_n$ are represented in terms of the matrices $(\mathbf{D}_0^1, \mathbf{D}_1^1), \ldots, (\mathbf{D}_0^n, \mathbf{D}_1^n)$, then their superposition M is also a MAP given by*

$$\mathbf{D}_0 = \mathbf{D}_0^1 \oplus \cdots \oplus \mathbf{D}_0^n \textit{ and } \mathbf{D}_1 = \mathbf{D}_1^1 \oplus \cdots \oplus \mathbf{D}_1^n \ .$$

With abuse of notation we use the same operator $\oplus$ for the MAPs' superposition, i.e.,

$$M = M_1 \oplus \cdots \oplus M_n \ .$$

Consider for example $M_3 = M_1 \oplus M_2$ for the MAPs from Figure 5, and in particular the corresponding matrices of hidden transitions

$$\mathbf{D_0^1} = \begin{pmatrix} -\lambda_1 & 0 \\ \mu_1 & -\mu_1 \end{pmatrix}, \ \mathbf{D_0^2} = \begin{pmatrix} -\lambda_2 & 0 \\ \mu_2 & -\mu_2 \end{pmatrix} .$$

Then the Kronecker sum $\mathbf{D_0^1} \oplus \mathbf{D_0^2}$ can be written as

$$\begin{pmatrix} -\lambda_1 & 0 & 0 & 0 \\ 0 & -\lambda_1 & 0 & 0 \\ \mu_1 & 0 & -\mu_1 & 0 \\ 0 & \mu_1 & 0 & -\mu_1 \end{pmatrix} + \begin{pmatrix} -\lambda_2 & 0 & 0 & 0 \\ \mu_2 & -\mu_2 & 0 & 0 \\ 0 & 0 & -\lambda_2 & 0 \\ 0 & 0 & \mu_2 & -\mu_2 \end{pmatrix} .$$

It is instructive to observe that the state-space of the Kronecker sum corresponds to the Cartesian product of the state spaces of M_1 and M_2 in lexicographical order, and which retains the Markovian properties of the (independent) superposed MAPs. Every state in M_1 corresponds to a block of 2 (i.e., the dimensionality of M_2) states in $M_1 \oplus M_2$; moreover, every state within such a block corresponds to a state in M_2. For the MAP M_3 from Figure 5.(c), the corresponding states are, in order, $\overline{C_1}\overline{C_2}$, $\overline{C_1}C_2$, $C_1\overline{C_2}$, and C_1C_2.

5.2.2. Input-Output: Σ, $\mathcal{R}$, and $\min(\Sigma, \mathcal{R})$ Caches

Let M be the MAP of cache requests and T be the TTL's PH distribution. We now prove that the input/miss process M', denoted formally using the notation $M' := M \oslash T$ is also a MAP, for all Σ, $\mathcal{R}$, and $\min(\Sigma, \mathcal{R})$ caches. Note that, unlike in Section 4 where the $\mathcal{R}$ model was simpler than the Σ model, the opposite holds for MAPs for which reason we start with Σ.

Theorem 3 (*MAP-PH-*Σ Cache (proof in Appendix 9.1))**.** *Consider a Σ-cache where requests arrive according to a MAP $M = (\mathbf{D}_0, \mathbf{D}_1)$. The TTLs are iid with a PH-distribution T and generator $\mathbf{P}$; also, M and T are independent. Then $M' := M \oslash T$ is a MAP with*

$$\mathbf{D}_0' = (\mathbf{P} \oplus \mathbf{D}_0) + \begin{pmatrix} \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_1 & \ddots & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{D}_1 \end{pmatrix} \quad \textit{and} \quad \mathbf{D}_1' = \begin{pmatrix} \mathbf{0} & \pi_1\mathbf{D}_1 & \pi_2\mathbf{D}_1 & \dots & \pi_m\mathbf{D}_1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{pmatrix},$$

where the $\mathbf{0}$ vectors have dimension $n \times n$. If M has n states and T has m transient and one absorbing states, then $\mathbf{D}_0'$ and $\mathbf{D}_1'$ are $n(m+1) \times n(m+1)$ matrices.

Constructing the output for a $\mathcal{R}$ cache follows along the same lines except that the state of the TTL is reset with each arrival while the object is in the cache. This difference is modelled explicitly in the second term of $\mathbf{D_0'}$ in the following theorem.

Theorem 4 (*MAP-PH-*$\mathcal{R}$ Cache (proof in Appendix 9.1))**.** *Under the same conditions as in Theorem 3, but for a $\mathcal{R}$ cache, $M' := M \oslash T$ is a MAP with*

$$\mathbf{D}_0' = (\mathbf{P} \oplus \mathbf{D}_0) + \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \pi_1\mathbf{D}_1 & \pi_2\mathbf{D}_1 & \dots & \pi_m\mathbf{D}_1 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \mathbf{0} & \pi_1\mathbf{D}_1 & \pi_2\mathbf{D}_1 & \dots & \pi_m\mathbf{D}_1 \end{pmatrix} \quad \textit{and} \quad \mathbf{D}_1' = \begin{pmatrix} \mathbf{0} & \pi_1\mathbf{D}_1 & \pi_2\mathbf{D}_1 & \dots & \pi_m\mathbf{D}_1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{pmatrix},$$

where the $\mathbf{0}$ vectors have dimension $n \times n$. If M has n states and T has m transient and one absorbing states, then $\mathbf{D}_0'$ and $\mathbf{D}_1'$ are $n(m+1) \times n(m+1)$ matrices.

Finally, the case of a $\min(\Sigma, \mathcal{R})$ cache exploits the known property that PH-distributions are closed under the minimum operator [10]):

Lemma 3 (Minimum of two PH-distributions)**.** *Let $T_1 = (\mathbf{S_1}, \pi_\mathbf{1})$ of order m, and $T_2 = (\mathbf{S_2}, \pi_\mathbf{2})$ of order q be two PH distributions. Then $\min(T_1, T_2)$ is a PH distribution of order mq, and given by $(\mathbf{S}, \pi)$ where*

$$\mathbf{S} = \mathbf{S_1} \oplus \mathbf{S_2} \ \textit{and} \ \pi = \pi_\mathbf{1} \otimes \pi_\mathbf{2} \ .$$

The construction of the $\min(\Sigma, \mathcal{R})$ output next follows by leveraging the property that the minimum of the two stopping times corresponding to Σ and $\mathcal{R}$, respectively, carries over to the TTLs' PH representations. In fact, given the minimum PH distribution from Lemma 3, the construction of the new matrix $\mathbf{D_0'}$ is comparable to the one from Theorem 4, and follows by repeating its construction for the second term of $\mathbf{D_0'}$.

Theorem 5 (*MAP-PH-*$\min(\Sigma, \mathcal{R})$ Cache (proof in Appendix 9.1))**.** *Consider a $\min(\Sigma, \mathcal{R})$ cache where requests arrive according to a MAP $M = (\mathbf{D}_0, \mathbf{D}_1)$. The two TTLs are iid with PH distributions T^Σ and $T^\mathcal{R}$ for Σ and $\mathcal{R}$, respectively. Arrivals and both TTLs are independent. If $\mathbf{P}$ denotes the generator of the PH distribution $\min(T^\Sigma, T^\mathcal{R})$ with corresponding initial vector $\pi = \pi^\mathbf{\Sigma} \otimes \pi^\mathcal{R}$ (see Lemma 3, where $\pi^\mathbf{\Sigma}$ and $\pi^\mathcal{R}$ are the initial vectors of T^Σ and $T^\mathcal{R}$, respectively), then $M' := M \oslash \min(T^\Sigma, T^\mathcal{R})$ is a MAP with*

$$\mathbf{D}_0' = (\mathbf{P} \oplus \mathbf{D}_0) + \begin{pmatrix} \mathbf{0}_{n\times n} & \mathbf{0}_{n\times nq} & \mathbf{0}_{n\times nq} & \dots & \mathbf{0}_{n\times nq} \\ \mathbf{0}_{nq\times n} & \mathbf{\Omega} & \mathbf{0}_{nq\times nq} & \dots & \mathbf{0}_{nq\times nq} \\ & \mathbf{0}_{nq\times nq} & \mathbf{\Omega} & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0}_{nq\times nq} \\ \mathbf{0}_{nq\times n} & \mathbf{0}_{nq\times nq} & & \dots & \mathbf{\Omega} \end{pmatrix}, \mathbf{D}_1' = \begin{pmatrix} \mathbf{0}_{n\times n} & \pi_1\mathbf{D}_1 & \pi_2\mathbf{D}_1 & \dots & \pi_{mq}\mathbf{D}_1 \\ \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} & \dots & \dots & \mathbf{0}_{n\times n} \\ \vdots & \vdots & & & \vdots \\ \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} & \dots & \dots & \mathbf{0}_{n\times n} \end{pmatrix}$$

$$\textit{and}\ \mathbf{\Omega} \ = \ \begin{pmatrix} {\pi^\mathcal{R}}_1\mathbf{D}_1 & \dots & {\pi^\mathcal{R}}_q\mathbf{D}_1 \\ \vdots & & \vdots \\ {\pi^\mathcal{R}}_1\mathbf{D}_1 & \dots & {\pi^\mathcal{R}}_q\mathbf{D}_1 \end{pmatrix},$$

where the $\mathbf{0}_{a\times b}$ vectors have dimension $a \times b$ and Ω has dimension $(nq \times nq)$, if M has n states, T^Σ has m transient states, and $T^\mathcal{R}$ has q transient states. $\mathbf{D}_0'$ and $\mathbf{D}_1'$ are $n(m\,q+1) \times n(m\,q+1)$ matrices.

We make the important observation that because the order of T_1 and T_2 from Lemma 3 matters for the order of the states in the Markov chain of the corresponding minimum, the order of $\min(T^{\Sigma}, T^{\mathcal{R}})$ cannot be interchanged without changing the structure of $\mathbf{D_0'}$.

The previous results (Theorems 3, 4, and 5) immediately extend to feedforward cache networks by using the closure of MAPs under thinning (see Nielsen [39] and Appendix 11). Also, we state the following immediate result, which follows from Proposition 5 in [4] and gives the caching metrics for a MAP model.

Lemma 4 (Steady-State Metrics for a MAP Cache)**.** *For the $\mathcal{R}$, Σ, and $\min(\Sigma, \mathcal{R})$ caching policies, if the input process is an n-state MAP $M = (\mathbf{D_0}, \mathbf{D_1})$, with steady-state probability vector $\mathbf{p} = (p_1, \dots p_n)$, and the output process is an n'-state MAP $M' = (\mathbf{D_0'}, \mathbf{D_1'})$, with steady-state vector $\mathbf{p}' = (p_1', \dots, p_{n'}')$, the miss and hit probabilities, and the cache occupancy are given by*

$$M = \frac{\mathbf{p'D_1'1'}}{\mathbf{pD_11}}, \quad H = 1 - M, \quad \pi = \sum_{i=1}^{n'} p_i' \, 1_{\{state\ i\ \in \mathsf{IN}\}}$$

where $\mathbf{1}$ and $\mathbf{1'}$ are all-ones vectors of dimensions $n \times 1$ and $n' \times 1$, respectively, and IN *was defined in the proof of Theorem 3 (Appendix 9.1).*

Finally, we point out a key drawback of the superposition and the input-output operations for MAPs, i.e., the state-spaces of the involved MAPs increase multiplicatively in the number of caches and the number of states of the TTLs' PH distribution, respectively (cf. Theorems 2-5).

Lemma 5 (Scaling of State Space)**.** *Assume a complete binary tree of height h with 2^{h-1} arriving n-state MAPs. All nodes implement either $\mathcal{R}$ or Σ caches, with an m-state PH-distribution for the TTLs. Then, for a fixed object, the state space size for the exact analysis of the miss process scales as $n^{2^h} m^{2(2^h-1)}$.*

The proof follows by induction; in the case of $\min(\Sigma, \mathcal{R})$ caches, the space complexity is even higher.

6. Numerical Results

In this section we first highlight a fundamental advantage of the $\min(\Sigma, \mathcal{R})$ policy, relative to $\mathcal{R}$ and Σ. Then we illustrate the numerical inaccuracy of Poisson approximations in tree networks.

First we consider objects of different popularities (the 1st, 10th-, 100th-, and 1000th under a *Zipf*(0.85) popularity law) in a line network of $\mathcal{R}$, Σ, and $\min(\Sigma, \mathcal{R})$ caches. Using Eq. (5) (as in [14, 24, 8, 37, 43]) we provision each cache with 100 objects (out of 1000 in total), under Poisson arrivals with rate 1 at the first cache and Erlang(2) distributed TTLs with implicit parameters. An ideal policy would cache the 100 most popular objects at the first cache [35] and thus would not need to cache them again at any downstream cache. Figure 6 (left) shows that the $\mathcal{R}$ model behaves the closest to the ideal case since the most popular object's cache occupancy is reduced from almost 1 (at cache 1) to less than 0.4 (at cache 2). In contrast, the Σ model needs to frequently cache the same object in both caches and performs suboptimally due to the significant change of the cache occupancy at the second cache. The crucial insight concerning $\min(\Sigma, \mathcal{R})$ is that it significantly improves the occupancy of Σ at the second cache, while also qualitatively improving the consistency of $\mathcal{R}$, at the expense of only a slight decrease in the cumulative hit ratio (as shown in Figure 6 (right)). While this fundamental advantage is pronounced for only few popular objects, we point out that the top 5 popular objects account for almost 20% of the overall traffic (due to the Zipf Law).

Next we investigate the accuracy of the popular Poisson approximation of miss processes in cache networks [14, 31, 34, 44, 41, 37, 43, 26, 37]. Figure 7 shows the relative error in the hit ratio of the Poisson approximation, relative to exact MAP results. We consider a standard binary tree scenario (as in [14, 44, 37, 16, 17, 15]) with four levels (15 nodes in total) and 1000 objects. Popular objects experience a $16-32\%$ error at the fourth level, and the error reaches $35-37\%$ for unpopular objects. Since these substantial errors can occur in medium-sized networks, which occur, e.g., when sizing overlay cache networks [46], we conclude that the Poisson approximation may be highly misleading.

For the results presented here we used the iterative Bi-CGSTAB [47, 45] method with an ILUTH preconditioner as implemented with the Nsolve tool [11]. This allows for numerical accuracy in the order of $1e-15$ and evaluates the tree within $3s$ for any particular TTL parameter.

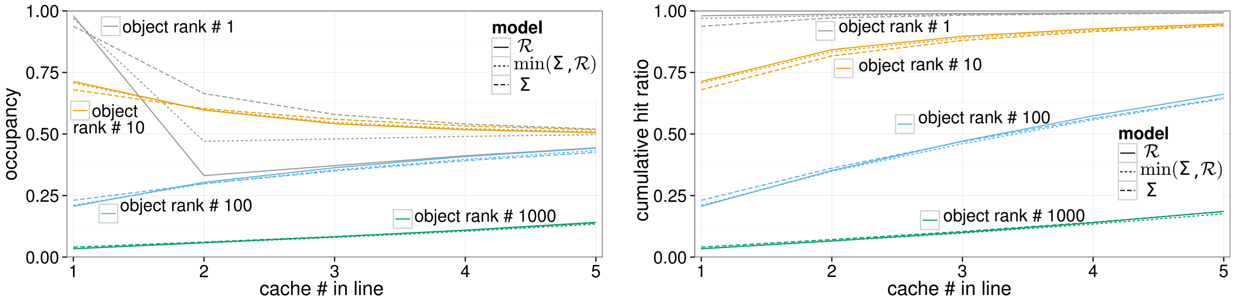


Figure 6: The cache occupancy (left) and cumulative hit ratio (right). For top popular objects, $\min(\Sigma, \mathcal{R})$ reconciles the cache occupancies of $\mathcal{R}$ with Σ at the downstream caches (see left), while improving the consistency of $\mathcal{R}$, and without significantly sacrificing the hit ratio (see right).

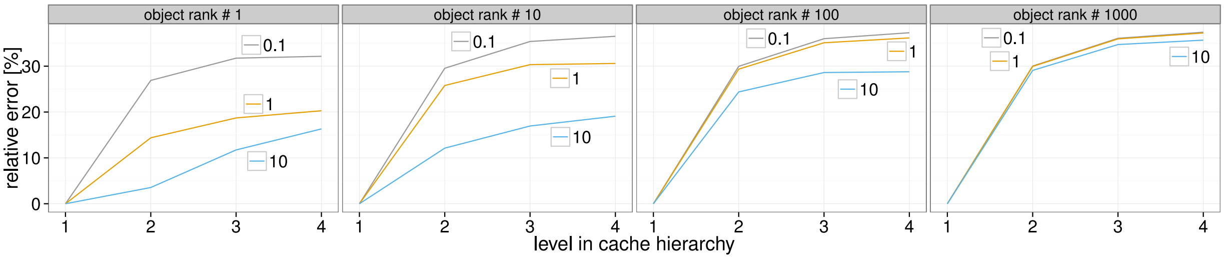


Figure 7: Relative error in the hit ratio of the Poisson approximation for a binary tree under three configurations (TTL-rate factors 0.1, 1, and 10) and four objects sampled from the popularity distribution.

7. Conclusion

In this paper we have provided the first exact analysis of TTL cache networks in great generality. We have developed two main methods covering three common TTL caching policies: $\mathcal{R}$, Σ, and $\min(\Sigma, \mathcal{R})$ which are employed in practical implementations. With the first method we have generalized existing results available for lines of $\mathcal{R}$ and Σ caches with renewal requests, by additionally accounting for the emerging $\min(\Sigma, \mathcal{R})$ policy which combines the benefits of $\mathcal{R}$ and Σ. The key idea was to conveniently formalize the three TTL caching policies by a stopped-sum representation, whose transform could thereafter be computed using a change of measure technique. To address the lack of closure of renewals under superposition (and hence the inherent limitation to line networks), our second method proposed to use the versatile class of MAPs to model cache non-renewal requests. The key contribution was to show that MAPs are closed under the input-output operation of all three caching policies, whereby TTLs follow PH distributions. This property was instrumental for the first exact analysis of feedforward TTL cache networks. While the method addressing MAPs has a much broader applicability, it suffers however from an exponential increase in the space complexity that numerical techniques can partly overcome. Overall, our results unequivocally and precisely capture the exact behavior of TTL caching networks, whereas existing Poisson approximations can be very misleading.

Acknowledgments

We thank Peter Buchholz from the University of Dortmund for his help and guidance in using the Nsolve tool to derive the numerical results presented in Section 6. We are also particularly grateful for the anonymous reviewers' helpful comments. This work was partially funded by the DFG grant Ci 195/1-1 and by the European Union under the EIT ICT Labs project SmartUC.

Appendix

8. Proofs from Section 4

8.1. Proof of Lemma 1

Let t_n denote the point process of the unsuccessful request times (i.e., the miss times) for $n \geq 1$ and $t_0 = 0$. Using the renewal property of $\{X_t\}_{t\geq 1}$ and $\{T_t\}_{t\geq 1}$, and the strong law of large numbers, we have

$$\lim_{n\to\infty} \frac{M(t_n)}{N(t_n)} = \lim_{n\to\infty} \frac{n}{\tau_1 + \cdots + \tau_n} = \frac{1}{E[\tau]} \ ,$$

where τ_i denotes the stationary sequence of stopping times, as defined in Eqs. (2) and (3), but starting from $t \geq t_{i-1}$ in the usual renewal sense. Moreover, since for $t \in (t_{i-1}, t_i]$

$$\frac{M(t_i) - 1}{N(t_i)} < \frac{M(t)}{N(t)} \leq \frac{M(t_{i-1}) + 1}{N(t_{i-1})} \ ,$$

the limit $\lim_{t\to\infty} \frac{M(t)}{N(t)}$ exists and Eq. (7) is proven. The particular expression for *G-G-*$\mathcal{R}$ cache follows directly from the geometric distribution of τ. □

8.2. Proof of Lemma 2

In the case of *G-G-*$\mathcal{R}$, denote the point process t_n of the request times, i.e., $t_n = \sum_{i=1}^n X_i$. Using the renewal property of $\{X_t\}_{t\geq 1}$ and $\{T_t\}_{t\geq 1}$, and the strong law of large numbers, we have

$$\lim_{n\to\infty} \frac{\int_0^{t_n} C(s)ds}{t_n} = \lim_{n\to\infty} \frac{\sum_{i=1}^n \min\{X_i, T_i\}}{\sum_{i=1}^n X_i} = \frac{E\left[\min\{X, T\}\right]}{E[X]} \ .$$

In the case of *G-G-*Σ, we use the same embedding t_n as in the proof of Lemma 1 such that

$$\lim_{n\to\infty} \frac{\int_0^{t_n} C(s)ds}{t_n} = \lim_{n\to\infty} \frac{\sum_{i=1}^n T_i}{\sum_{i=1}^n S_{\tau_i}} = \frac{E\left[T\right]}{E\left[S_\tau\right]} \ ,$$

where $S_{\tau_i} := X_{\tau_{i-1}+1} + \cdots + X_{\tau_{i-1}+\tau_i}$.

In both cases, the extensions of the limits to the whole line follows by a bounding argument as in the proof of Lemma 1. The proof for *G-G-*min$(\Sigma, \mathcal{R})$ follows using the same embedding points as for *G-G-*Σ.□

8.3. Proof of Theorem 1

Fix $T \geq 0$ and choose $Y := \mathcal{L}(X)^\tau 1_{\{\tau\leq T\}}$ which is $\mathcal{F}_T$ measurable. Applying the integration rule from Eq. (10) and the properties of conditional expectation we get

$$\begin{aligned}
\tilde{E}\left[\mathcal{L}(X)^\tau 1_{\{\tau\leq T\}}\right] &= E\left[\mathcal{L}(X)^\tau 1_{\{\tau\leq T\}} L_T\right] \\
&= E\left[E\left[\mathcal{L}(X)^\tau 1_{\{\tau\leq T\}} L_T \mid \mathcal{F}_\tau\right]\right] \\
&= E\left[\mathcal{L}(X)^\tau 1_{\{\tau\leq T\}} E\left[L_T \mid \mathcal{F}_\tau\right]\right] \\
&= E\left[e^{-\omega S_\tau} 1_{\{\tau\leq T\}}\right] \ .
\end{aligned}$$

In the last line we used the martingale property of L_T, i.e., $E\left[L_T \mid \mathcal{F}_\tau\right] = L_\tau$. From the monotonicity of $1_{\{\tau\leq T\}}$ in T, the proof is complete by applying Lebesgue's dominated convergence theorem (see Theorem 16.4 in Billingsley [9]). □

8.4. Proof of Proposition 1

Fix $t \geq 1$ and $x \geq 0$. The distribution $\tilde{F}(x)$ follows immediately from the integration rule from Eq. (10):

$$\tilde{\mathbb{P}}\left(X_t \leq x\right) = \int \frac{e^{-\omega X_t}}{\mathcal{L}(X)} 1_{\{X_t \leq x\}} d\mathbb{P}_{X_t} \ ,$$

where $\mathbb{P}_{X_t}$ is the projection of $\mathbb{P}$ on $\sigma(X_t)$. In turn, for $\tilde{T}(x)$, we have similarly

$$\begin{aligned}
\tilde{\mathbb{P}}\left(T_t \leq x\right) &= \int_{\Omega_{X_t} \times \{T_t \leq x\}} \frac{e^{-\omega X_t}}{\mathcal{L}(X)} d\mathbb{P}_{X_t} \times \mathbb{P}_{T_t} \\
&= \int_{T_t \leq x} \int_{\Omega_{X_t}} \frac{e^{-\omega X_t}}{\mathcal{L}(X)} d\mathbb{P}_{X_t} d\mathbb{P}_{T_t} \\
&= \int_{T_t \leq x} d\mathbb{P}_{T_t} = F_T(x) \ ,
\end{aligned}$$

where Ω_{X_t} denotes the (projected) sample space corresponding to X_t. In the first line we used the independence of X_t and T_t, i.e., the random vector (X_t, T_t) has the product measure $d\mathbb{P}_{X_t} \times \mathbb{P}_{T_t}$, where $\mathbb{P}_{X_t}$ and $\mathbb{P}_{T_t}$ are the projections of $\mathbb{P}$ on $\sigma(X_t)$ and $\sigma(T_t)$, respectively. In the second line we used Fubini's theorem.

Lastly, consider $B_1 \in \sigma(X_t)$ and $B_2 \in \sigma(T_t)$. Using again the independence of X_t and T_t (under $\mathbb{P}$) and Fubini's theorem we get

$$\begin{aligned}
\tilde{\mathbb{P}}\left(X_t \in B_1, T_t \in B_2\right) &= \int \frac{e^{-\omega X_t}}{\mathcal{L}(X)} 1_{\{X_t \in B_1, T_t \in B_2\}} d\mathbb{P} \\
&= \int_{\Omega_{X_t}} \frac{e^{-\omega X_t}}{\mathcal{L}(X)} 1_{\{X_t \in B_1\}} d\mathbb{P}_{X_t} \int_{T_t \in B_2} d\mathbb{P}_{T_t} \\
&= \tilde{\mathbb{P}}\left(X_t \in B_1\right) \tilde{\mathbb{P}}\left(T_t \in B_2\right) \ ,
\end{aligned}$$

which completes the proof. □

8.5. Proof of Corollary 1

Using the integration rule from Eq. (10) we first compute

$$\tilde{\mathbb{P}}(X \leq T) = \tilde{\mathbb{E}}\left[1_{\{X \leq T\}}\right] = \frac{\mathbb{E}\left[1_{\{X \leq T\}} e^{-\omega X}\right]}{\mathbb{E}\left[e^{-\omega X}\right]} = \frac{\psi(\omega)}{\mathcal{L}(X)} \ ,$$

such that the pmf of τ is

$$\tilde{\mathbb{P}}(\tau = t) = \tilde{\mathbb{P}}(X \leq T)^{t-1} \left(1 - \tilde{\mathbb{P}}(X \leq T)\right) = \left(\frac{\psi(\omega)}{\mathcal{L}(X)}\right)^{t-1} \left(1 - \frac{\psi(\omega)}{\mathcal{L}(X)}\right) \ .$$

Finally, applying Theorem 1 and manipulating progression series yields

$$\begin{aligned}
\mathbb{E}\left[e^{-\omega S_\tau}\right] &= \tilde{\mathbb{E}}\left[\mathcal{L}(X)^\tau\right] \\
&= \sum_{t=1}^{\infty} \mathcal{L}(X)^t \left(\frac{\psi(\omega)}{\mathcal{L}(X)}\right)^{t-1} \left(1 - \frac{\psi(\omega)}{\mathcal{L}(X)}\right) \\
&= \frac{\mathcal{L}(X) - \psi(\omega)}{1 - \psi(\omega)} \ ,
\end{aligned}$$

which completes the proof. □

8.6. Proof of Corollary 2

We first need to introduce the distribution convolution of S_t, for all $t \geq 1$, in the new space $(\Omega, \mathcal{F}, \tilde{\mathbb{P}})$. These are given for all $x \geq 0$ by by $\tilde{F}^1(x) := \tilde{F}(x)$ as in Proposition 1 and then recursively for $t > 1$ by the convolutions

$$\tilde{F}^t(x) = \int_0^x \tilde{F}^{t-1}(x-y) d\tilde{F}(y) \ ,$$

where $\tilde{F}^0(x) = 0$ for $x < 0$ and $\tilde{F}^0(x) = 1$ for $x \geq 0$. Assume also the existence of the corresponding densities $\tilde{f}^t$.

We use the pmf of τ in the tilted space $\tilde{\mathbb{P}}(\tau = t) = \tilde{\mathbb{P}}(S_t > T, \, S_{t-1} \leq T)$, condition on T, and then on $S_{t-1} = X_1 + \cdots + X_{t-1}$ and finally recall from Proposition 1 that $\tilde{g}(x) = g(x)$. This gives:

$$\begin{aligned}
\mathbb{E}\left[e^{-\omega S_\tau}\right] &= \tilde{\mathbb{E}}\left[\mathcal{L}(X)^\tau\right] = \sum_{t=1}^{\infty} \mathcal{L}(X)^t \tilde{\mathbb{P}}(\tau = t) \\
&= \sum_{t=1}^{\infty} \mathcal{L}(X)^t \int_0^\infty \tilde{\mathbb{P}}(S_t > x, \, S_{t-1} \leq x) \, \tilde{g}(x) \, dx \\
&= \sum_{t=1}^{\infty} \mathcal{L}(X)^t \int_0^\infty \int_0^x (1 - \tilde{F}(x-y)) \, \tilde{f}^{\,t-1}(y) \, dy \, g(x) \, dx \\
&= \sum_{t=1}^{\infty} \mathcal{L}(X)^t \int_0^\infty \left(\tilde{F}^{\,t-1}(x) \ - \ \tilde{F}^{\,t}(x)\right) g(x) \, dx \ . \qquad (14)
\end{aligned}$$

□

8.7. Equivalence to the expression in Fofack et al. [16]

We continue with the expression from Eq. (14) (which is in the form as stated in Corollary 2) and show how to derive the result by Fofack *et al.*

$$\begin{aligned}
\mathbb{E}\left[e^{-\omega S_\tau}\right] &= \sum_{t=1}^{\infty} \mathcal{L}(X)^t \int_0^\infty \left(\frac{\mathbb{E}\left[e^{-\omega S_{t-1}} 1_{\{S_{t-1} \leq x\}}\right]}{\mathcal{L}(X)^{t-1}} - \frac{\mathbb{E}\left[e^{-\omega S_t} 1_{\{S_t \leq x\}}\right]}{\mathcal{L}(X)^t} \right) g(x) \, dx \\
&= \sum_{t=1}^{\infty} \int_0^\infty \mathcal{L}(X) \mathbb{E}\left[e^{-\omega S_{t-1}} 1_{\{S_{t-1} \leq x\}}\right] g(x) \, dx - \sum_{t=1}^{\infty} \int_0^\infty \mathbb{E}\left[e^{-\omega S_t} 1_{\{S_t \leq x\}}\right] g(x) \, dx \qquad (15)
\end{aligned}$$

We start by considering the first term, in which we use Fubini's theorem to exchange the order of the integrals in the second step

$$\begin{aligned}
\sum_{t=1}^{\infty} \int_0^\infty \mathbb{E}\left[e^{-\omega S_t} 1_{\{S_t \leq x\}}\right] g(x) \, dx &= \sum_{t=1}^{\infty} \int_0^\infty \int_0^\infty e^{-\omega y} 1_{\{y \leq x\}} f^t(y) g(x) \, dy \, dx \\
&= \sum_{t=1}^{\infty} \int_0^\infty e^{-\omega y} (1 - G(y)) f^t(y) \, dy \\
&= \Phi(\omega) \ ,
\end{aligned}$$

where $\Phi(\omega) := \sum_{t=1}^{\infty} \int_0^{\infty} e^{-\omega y} f^t(y)(1 - G(y))dy$. We similarly proceed for the second term:

$$\begin{aligned}
\sum_{t=1}^{\infty} \int_0^{\infty} \mathbb{E}\left[e^{-\omega S_{t-1}} 1_{\{S_{t-1} \leq x\}}\right] g(x)\, dx &= \sum_{t=1}^{\infty} \int_0^{\infty} \int_0^{\infty} e^{-\omega y} 1_{\{y \leq x\}} f^{t-1}(y) g(x)\, dy\, dx \\
&= \sum_{t=0}^{\infty} \int_0^{\infty} \int_0^{\infty} e^{-\omega y} 1_{\{y \leq x\}} f^{t}(y) g(x)\, dy\, dx \\
&= \sum_{t=0}^{\infty} \int_0^{\infty} e^{-\omega y} f^t(y)(1 - G(y))\, dy \\
&= \int_0^{\infty} e^{-\omega y} f^0(y)(1 - G(y))\, dy + \Phi(\omega) \\
&= 1 + \Phi(\omega)\ . \qquad \text{(because } f^0(y) = 1_{\{y=0\}}\text{)}
\end{aligned}$$

Finally, we substitute the two terms back into Eq. (15) and obtain the same expression as in Proposition 2 from [16]: $\mathbb{E}\left[e^{-\omega S_\tau}\right] = \mathcal{L}(X)(1 + \Phi(\omega)) - \Phi(\omega)$. □

8.8. Corollary for the G-G-min($\mathcal{R}, \Sigma$) case

Corollary 3 (G-G-min$(\mathcal{R}, \Sigma)$)**.**
Let τ as in Eq. (4), $g(\cdot)$ the density of T^{Σ}, $H(\cdot)$ the distribution of $T^{\mathcal{R}}$, and $\psi(\omega)$ as in Corollary 1. Then, for some $\omega > 0$ the Laplace transform of the inter-miss time in the G-G-min($\mathcal{R}, \Sigma$) model is given by

$$\mathcal{L}(S_\tau) = \Phi_1(\omega)(2\mathcal{L}(X_1) - \psi(\omega)) + \Phi_2(\omega)\ ,$$

where $\Phi_1(\omega) := \sum_{t \geq 1} \int_0^{\infty} \mathbb{E}\left[e^{-\omega S_t} 1_{\{S_t \leq k\}} \prod_{i=1}^{t} (1 - H(X_i))\right] g(k)dk$ and
$\Phi_2(\omega) := \sum_{t \geq 1} \int_0^{\infty} \mathbb{E}\left[e^{-\omega S_t} 1_{\{S_t \leq k\}} \prod_{i=1}^{t-1} (1 - H(X_i))\right] g(k)dk$.

Proof. The proof follows along the same lines as the one in Appendix 8.6, using the tilted pmf of τ

$$\begin{aligned}
\tilde{\mathbb{P}}(\tau = t) =& \tilde{\mathbb{P}}(\forall_{i<t} X_i \leq T_i^{\mathcal{R}}, X_t > T_t^{\mathcal{R}}, S_{t-1} \leq T_1^{\Sigma}, S_t > T_1^{\Sigma}) \\
&+ \tilde{\mathbb{P}}(\forall_{i<t} X_i \leq T_i^{\mathcal{R}}, X_t > T_t^{\mathcal{R}}, S_t \leq T_1^{\Sigma}) + \tilde{\mathbb{P}}(\forall_{i \leq t} X_i \leq T_i^{\mathcal{R}}, S_{t-1} \leq T_1^{\Sigma}, S_t > T_1^{\Sigma})\ .
\end{aligned}$$

An explicit solution follows by conditioning on T_1^{Σ}, then on $X_1 \ldots X_t$, and finally on $T_1^{\mathcal{R}} \ldots T_t^{\mathcal{R}}$:

$$\begin{aligned}
\mathbb{E}\left[e^{-\omega\, S_\tau}\right] =& \sum_{t \geq 1} \left(2 \int_0^{\infty} \mathbb{E}\left[e^{-\omega\, S_{t-1}} 1_{\{S_{t-1} \leq k\}} \prod_{i=1}^{t-1} (1 - H(X_i)) \right] (\mathcal{L}(X_t) - \psi(\omega)) g(k) dk \right. \\
&- \int_0^{\infty} \mathbb{E}\left[e^{-\omega\, S_t} 1_{\{S_t \leq k\}} \prod_{i=1}^{t-1} (1 - H(X_i)) H(X_t) \right] g(k) dk \\
&+ \left. \int_0^{\infty} \mathbb{E}\left[e^{-\omega\, S_{t-1}} 1_{\{S_{t-1} \leq k\}}\right] \prod_{i=1}^{t-1} (1 - H(X_i)) \psi(\omega) - \mathbb{E}\left[e^{-\omega\, S_t} 1_{\{S_t \leq k\}} \prod_{i=1}^{t} (1 - H(X_i)) \right] g(k) dk \right)\ .
\end{aligned}$$

Rewriting with Φ_1 and Φ_2 leads to the term used in the Corollary. □

9. Proofs from Section 5

9.1. Proof of Theorem 3

First, it is easy to check that M' is a MAP according to Definition 6. The state space of M' is the Cartesian product of the state spaces of T and M (thus the term $\mathbf{P} \oplus \mathbf{D}_0$ in the expression of $\mathbf{D}_0'$). Note that, for technical reasons, the order of T and M in the Cartesian product is the opposite to the order in

$M \oslash T$. The Cartesian product accounts for all the combinations of states from T and M. In particular, every state in T corresponds to a block of n states in M' (e.g., the first n rows and columns in $\mathbf{D}_0'$ and $\mathbf{D}_1'$), each corresponding to a state in M (recall the example after Theorem 2); moreover, block i corresponds to the states $(i-1)n + j \ \forall j = 1, \ldots, n$.

Next, to prove that M' models the miss process, we divide the $m+1$ blocks of M' into two groups: OUT and IN. The OUT group accounts for the situation when the object is 'out of the cache' and corresponds to the absorbing state of T, i.e., when the TTL is expired. While in any of the OUT states (corresponding to a position (i,j) in $\mathbf{D}_0'$ and $\mathbf{D}_1'$ with $1 \leq i \leq n$ and $1 \leq j \leq n(m+1)$), there are both hidden transitions (only due to the second Kronecker product in $\mathbf{P} \oplus \mathbf{D}_0$; the first product does not contribute because the current state of T is absorbing according to our representation of $\mathbf{P}$ from Definition 7) and active transitions (see the first block of rows in $\mathbf{D}_1'$). An active transition regenerates the phase of the TTL according to the stationary distribution π and consequently M' jumps to an IN block.

The IN group accounts for the situation when the object is 'in the cache', and each block within corresponds to one of the phases of T. While in any of the IN states (corresponding to a position (i,j) in $\mathbf{D}_0'$ and $\mathbf{D}_1'$ with $(n+1) \leq i \leq n(m+1)$ and $1 \leq j \leq n(m+1)$) there are only hidden transitions. Some are given by the entries of $\mathbf{P} \oplus \mathbf{D}_0$, and thus modelling the joint evolution of M and T. Importantly, we remark that since M' is within an IN group, the active transitions from $\mathbf{D}_1$ become passive; this is expressed in the second term of $\mathbf{D}_0'$. Moreover, the time between any two consecutive such transformed passive transitions corresponds to an element X_s from the definition of the stopping time of a Σ-cache (recall Eq. (3)). Finally, M' eventually jumps to the OUT block when an exit transition from $\mathbf{T}$ occurs.

Note that the proof implicitly uses the fact that the superposition of independent MAPs retains the underlying Markovian properties. □

9.2. Proof of Theorem 4

The proof is identical to the previous one, except for accounting for the difference between Σ and $\mathcal{R}$ caches (see Eq. (3) vs. Eq. (2)). Concretely, while in the states of the IN group, an active transition becomes passive (as in the Σ case), but it also resets the phase of the TTL according to the probability vector π. □

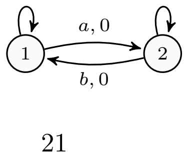
9.3. Proof of Theorem 5

The eviction event for the minimum of the two stopping times as defined in Eq. (4) translates into either reaching the accepting state of T^{Σ}, or reaching the accepting state of $T^{\mathcal{R}}$ without an intermittent arrival. The minimum distribution of T^{Σ} and $T^{\mathcal{R}}$ captures this behavior up to the resetting of $T^{\mathcal{R}}$ upon arrivals. As mentioned in the proof of Theorem 4, an arrival resets $T^{\mathcal{R}}$ back to its initial state defined by its initial vector $\pi^{\mathcal{R}}$; however, the state of T^{Σ} is preserved. By Lemma 3 (and the underlying Kronecker sum), the states in $\min(\Sigma, \mathcal{R})$ are lexicographically ordered with the states of Σ followed by the states of $\mathcal{R}$. According to this order and because the reset behavior only changes the state of $T^{\mathcal{R}}$, an arrivals' effect remains local to each diagonal block Ω. Each Ω corresponds to the $\mathcal{R}$ reset matrix, as defined in the second term of $\mathbf{D_0'}$ in Theorem 4. □

10. Examples for Caches with PH TTLs

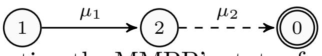
The purpose of this section is to build on the intuitive idea of IN and OUT sets of states, which occured in the proofs in Section 9. The examples given here address more complex MAPs as they arrive from caches having TTLs of phase type.

We will give examples for the application of each of the Theorems 3, 4, and 5. For all three cache models (Σ, $\mathcal{R}$, $\min(\Sigma, \mathcal{R})$), we assume an MMPP request model denoted by M, in the following reproduction of Figure 3:

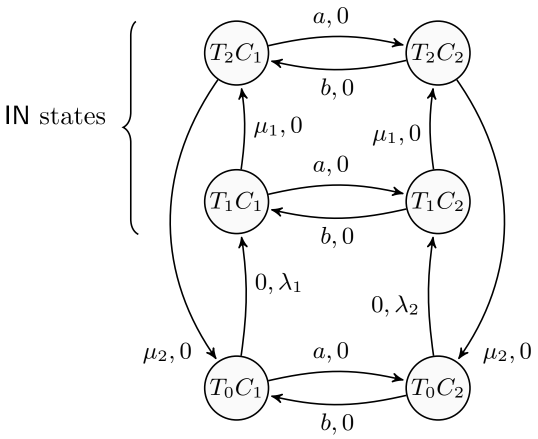


10.1. Σ Cache Model

We assume the following TTL T (in Markov chain representation and which starts in state 1 with probability one).



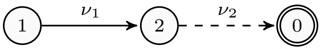
The output MAP is constructed by replicating the MMPP's states for each state of the TTL and adjusting for the inherent cache property, that no misses occur while the object is in the cache. This basic idea is reflected in taking the Cartesian product of T and M and subsequently making all of $\mathbf{D_1}$'s transitions passive (cf. definition of $\mathbf{D'_0}$ in Theorem 3). We denote each state by (T_iC_j), where i represents the active state of the TTL and j is the active state of the MMPP request process.



Note that we do not draw self-loops unless they are active transitions (i.e., the entries $D'_{0_{ii}}$ are not drawn, whereas an entry $D'_{1_{ii}}$ is drawn, as in the MMPP example). While the cache is in the IN state, further arrivals do not change the state of the cache. Thus, there are no transitions with λ_1 or λ_2 in the IN part of the resulting cache.

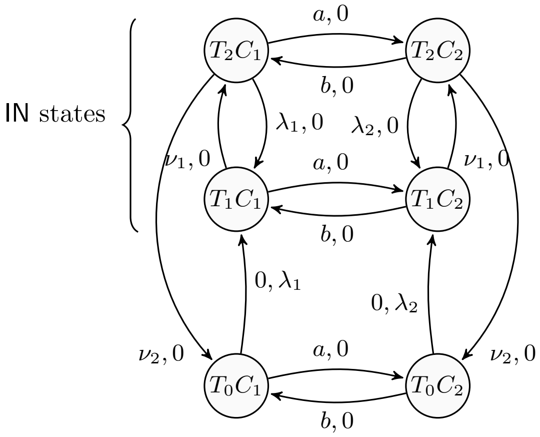
10.2. $\mathcal{R}$ Cache Model

Similarly to the Σ case we consider the MMPP arrival process M and the following TTL T (in Markov chain representation and which starts in state 1 with probability one):



Constructing the output MAP for the $\mathcal{R}$ case bears a subtle difference from the Σ case. The basic idea is again to replicate the MMPP's states for each state of the TTL but then we have to accommodate for the $\mathcal{R}$ resetting behavior of this cache model: every arrival while the object is in the cache resets the TTL's state according to its initial vector.

Recalling the notations from Theorem 4, this idea is reflected by taking the Cartesian product of T and M and subsequently adjusting for the "resetting behavior", i.e., by making $\mathbf{D_1}$'s transitions passive and resetting T's state. We again denote each state by (T_iC_j), where i represents the active state of the TTL and j is the active state of the MMPP request process.

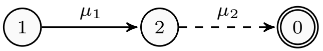


The difference to the Σ output MAP are additional edges $(T_2C_1 \to T_1C_1)$ and $(T_2C_2 \to T_1C_2)$, which model the inherent reset behavior of each arrival in the $\mathcal{R}$ model. Also note that if there was a greater number of TTL states and a non-trivial initial probability vector π for T, then the passive transitions '$\lambda_1, 0$' and '$\lambda_2, 0$' for each state of T would be directed to the initial states according to π and independently of T's current state. This is represented by the second term of $\mathbf{D_0'}$ in Theorem 4 by repeating the row with $\pi_i \mathbf{D_1}$ for each state of the TTL.

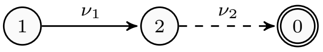
Finally, we turn to the $\min(\Sigma, \mathcal{R})$ cache model which is more complicated due to the higher number of states involved.

10.3. $\min(\Sigma, \mathcal{R})$ *Cache Model*

Consider the same MMPP arrival process M with the following TTL representations. For the Σ part of the model, the TTL is called T^{Σ}:

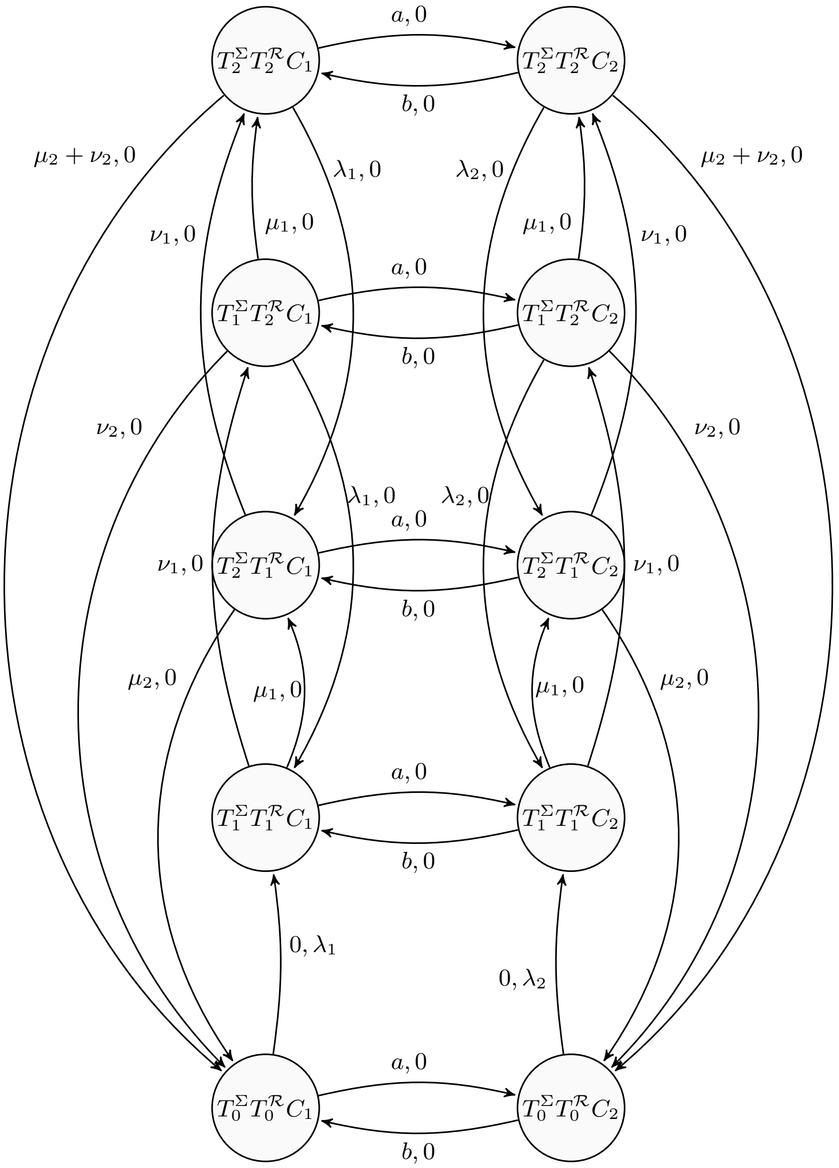


In turn, for the $\mathcal{R}$ part of the model, the TTL is called $T^{\mathcal{R}}$:



The corresponding output for a $\min(\Sigma, \mathcal{R})$ cache follows by constructing the PH minimum for $\min(T^{\Sigma}, T^{\mathcal{R}})$ which has four transient states and one absorbing state: $(T_1^{\Sigma}T_1^{\mathcal{R}})$, $(T_2^{\Sigma}T_1^{\mathcal{R}})$, $(T_1^{\Sigma}T_2^{\mathcal{R}})$, $(T_2^{\Sigma}T_2^{\mathcal{R}})$, and $(T_0^{\Sigma}T_0^{\mathcal{R}})$.

Then, we replicate the MMPP's states for each state of $\min(T^{\Sigma}, T^{\mathcal{R}})$ and link this Cartesian product construction with the reset behavior of the $\mathcal{R}$ model. As pointed out in the proof of Theorem 5, the resetting behavior of $\mathcal{R}$ has to preserve the state of T^{Σ}. This behavior is represented in the following output MAP by the two '$\lambda_1, 0$' and the two '$\lambda_2, 0$' transitions. We denote each state by $(T_i^{\Sigma}T_j^{\mathcal{R}}C_k)$, where i represents the active state of the Σ-TTL, j the active state of the $\mathcal{R}$-TTL, and k the active state of the MMPP request process.



$T_2^\Sigma T_2^\mathcal{R} C_1$
$a, 0$
$T_2^\Sigma T_2^\mathcal{R} C_2$
$b, 0$
$\mu_2 + \nu_2, 0$
$\lambda_1, 0$
$\lambda_2, 0$
$\mu_2 + \nu_2, 0$
$\nu_1, 0$
$\mu_1, 0$
$\mu_1, 0$
$\nu_1, 0$
$T_1^\Sigma T_2^\mathcal{R} C_1$
$a, 0$
$T_1^\Sigma T_2^\mathcal{R} C_2$
$b, 0$
$\nu_2, 0$
$\nu_2, 0$
$\lambda_1, 0$
$\lambda_2, 0$
$a, 0$
$\nu_1, 0$
$T_2^\Sigma T_1^\mathcal{R} C_1$
$T_2^\Sigma T_1^\mathcal{R} C_2$
$\nu_1, 0$
$b, 0$
$\mu_2, 0$
$\mu_1, 0$
$\mu_1, 0$
$\mu_2, 0$
$a, 0$
$T_1^\Sigma T_1^\mathcal{R} C_1$
$T_1^\Sigma T_1^\mathcal{R} C_2$
$b, 0$
$0, \lambda_1$
$0, \lambda_2$
$a, 0$
$T_0^\Sigma T_0^\mathcal{R} C_1$
$T_0^\Sigma T_0^\mathcal{R} C_2$
$b, 0$

11. Probabilistic Splitting of Arrivals

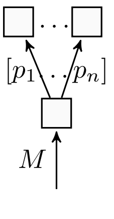


Figure 8: n-fold split of M

Apart from the superposition and the input-output operations, a third operation is called *splitting* and allows to split the MAP of an input (output) process, as shown in Figure 8. This works as an inverse operation of the superposition operator and allows to model the behavior of a cache feedforward network.

A common splitting operation is when the process is split accordingly to some fixed probabilities. Such a construction allows to capture the behavior of an idealized load balancer.

Lemma 6 (Splitting)**.** *Assume a MAP $M = (\mathbf{D_0}, \mathbf{D_1})$ is split into n sub processes according to a stochastic n-vector p. The resulting processes are characterized by the MAPs $M_i = (\mathbf{D_0^i}, \mathbf{D_1^i})$, where*

$$\mathbf{D_0^i} = \mathbf{D_0} + (1 - p_i)\mathbf{D_1} \text{ and } \mathbf{D_1^i} = p_i \mathbf{D_1} \ , \quad \text{for } 1 \leq i \leq n \ .$$

Proof. This is an extension of the known result that a single MAP is closed under thinning [39]. □

We point out that an input process represented as a MAP can be split in different ways of which many can be captured by a thinning operation. As further examples, the MAP arrival model is also closed under splitting requests according to their origin, or more generally, when the splitting decisions can be described by a Markov process. Besides accounting for splitting operations, our results can be further extended to account for various cache replication strategies as considered in Martina *et al.* [37].

References

[1] Squid Web Cache FAQ. `http://wiki.squid-cache.org/SquidFaq/InnerWorkings`. acc. 2014-04-05.

[2] OpenFlow Switch Specification 1.4.0, October 2013.

[3] Amazon Web Service. *Amazon ElastiCache User Guide*, API version 2013-06-15 edition.

[4] S. Asmussen. Matrix-analytic models and their analysis. *Scandinavian Journal of Statistics*, 27(2):193–226, June 2000.

[5] S. Asmussen. *Applied probability and queues*, volume 2. Springer, 2003.

[6] S. Asmussen and G. Koole. Marked point processes as limits of Markovian arrival streams. *Journal of Applied Probability*, 30(2):365–372, June 1993.

[7] O. Bahat and A. M. Makowski. Measuring consistency in TTL-based caches. *Performance Evaluation*, 62(1):439–455, 2005.

[8] G. Bianchi, A. Detti, A. Caponi, and N. Blefari Melazzi. Check before storing: what is the performance price of content integrity verification in LRU caching? *ACM SIGCOMM Computer Communication Review*, 43(3):59–67, 2013.

[9] P. Billingsley. *Probability and Measure.* Wiley, 3rd edition, 1995.

[10] L. Breuer and D. Baum. *An introduction to queueing theory and matrix-analytic methods.* Springer, 2005.

[11] P. Buchholz. The Nsolve Program. Technical report, University of Dortmund, 2010. `http://ls4-www.cs.tu-dortmund.de/download/buchholz/struct-matrix-market.html` acc. 2014-04-05.

[12] G. Casale. Tutorial: Building accurate workload models using markovian arrival processes. In *Proceedings of ACM SIGMETRICS*, pages 357–358, 2011. available `http://www.sigmetrics.org/sigmetrics2011/tutorials/tutorial1.pdf`.

[13] G. Casale, N. Mi, and E. Smirni. Bound analysis of closed queueing networks with workload burstiness. In *Proceedings of ACM SIGMETRICS*, pages 13–24, 2008.

[14] 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.

[15] N. Choungmo Fofack, P. Nain, G. Neglia, and D. Towsley. Performance evaluation of hierarchical TTL-based cache networks. *Computer Networks*, 65:212–231, 2014.

[16] N. E. Choungmo Fofack and S. Alouf. Modeling modern DNS caches. In *Proceedings of IEEE VALUETOOLS*, 2013.

[17] N. E. Choungmo Fofack, D. Towsley, M. Badov, M. Dehghan, and D. L. Goeckel. An approximate analysis of heterogeneous and general cache networks. Rapport de recherche RR-8516, INRIA, Apr. 2014. available http://hal.inria.fr/hal-00975339/PDF/RR-8516.pdf.

[18] E. Cohen, E. Halperin, and H. Kaplan. Performance aspects of distributed caches using TTL-based consistency. In *Automata, Languages and Programming*, pages 744–756. Springer, 2001.

[19] E. Cohen and H. Kaplan. Aging through cascaded caches: Performance issues in the distribution of web content. In *Proceedings of ACM SIGCOMM*, pages 41–53, 2001.

[20] S. H. Cox, Y. Lin, and S. Wang. Multivariate exponential tilting and pricing implications for mortality securitization. *Journal of Risk and Insurance*, 73(4):719–736, 2006.

[21] A. Dan and D. Towsley. An approximate analysis of the LRU and FIFO buffer replacement schemes. In *Proceedings of ACM SIGMETRICS*, pages 143–152, 1990.

[22] A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications.* Springer, 2nd edition, 1998.

[23] R. Fagin and T. G. Price. Efficient calculation of expected miss ratios in the independent reference model. *SIAM Journal on Computing*, 7(3):288–297, 1978.

[24] C. Fricker, P. Robert, and J. Roberts. A versatile and accurate approximation for LRU cache performance. In *Proceedings of ITC*, pages 1–8, 2012.

[25] C. Fricker, P. Robert, J. Roberts, and N. Sbihi. Impact of traffic mix on caching performance in a content-centric network. In *IEEE NOMEN Workshop on Emerging Design Choices in Name-Oriented Networking*, pages 310–315, 2012.

[26] M. Gallo, B. Kauffmann, L. Muscariello, A. Simonian, and C. Tanguy. Performance evaluation of the random replacement policy for networks of caches. In *Proceedings of ACM SIGMETRICS/ PERFORMANCE*, pages 395–396, 2012.

[27] E. Gelenbe. A unified approach to the evaluation of a class of replacement algorithms. *IEEE Transactions on Computers*, 100(6):611–618, 1973.

[28] A. Gut. *Stopped Random Walks: Limit Theorems and Applications*. Springer, 2009.

[29] Y. T. Hou, J. Pan, B. Li, and S. S. Panwar. On expiration-based hierarchical caching systems. *IEEE Journal on Selected Areas in Communications*, 22(1):134–150, 2004.

[30] P. R. Jelenkovic. Asymptotic approximation of the move-to-front search cost distribution and least-recently used caching fault probabilities. *The Annals of Applied Probability*, 9(2):430–464, 1999.

[31] P. R. Jelenković and X. Kang. Characterizing the miss sequence of the LRU cache. *ACM SIGMETRICS Performance Evaluation Review*, 36(2):119–121, 2008.

[32] J. Jung, A. W. Berger, and H. Balakrishnan. Modeling TTL-based internet caches. In *Proceedings of IEEE INFOCOM*, pages 417–426, 2003.

[33] W. F. King III. Analysis of demand paging algorithms. In *IFIP Congress (1)*, pages 485–490, 1971.

[34] N. Laoutaris, H. Che, and I. Stavrakakis. The LCD interconnection of LRU caches and its analysis. *Performance Evaluation*, 63(7):609–634, 2006.

[35] Z. Liu, P. Nain, N. Niclausse, and D. Towsley. Static caching of web servers. In *Proceedings of SPIE*, pages 179–190, 1997.

[36] D. M. Lucantoni, K. S. Meier-Hellstern, and M. F. Neuts. A single-server queue with server vacations and a class of non-renewal arrival processes. *Advances in Applied Probability*, 22(3):676–705, Sept. 1990.

[37] V. Martina, M. Garetto, and E. Leonardi. A unified approach to the performance analysis of caching systems. In *Proceedings of IEEE INFOCOM*, 2014.

[38] M. Musiela and M. Rutkowski. *Martingale methods in financial modelling*. Springer, 2005.

[39] B. F. Nielsen. Note on the Markovian arrival process. 1998. `http://www2.imm.dtu.dk/courses/04441/map.pdf`.

[40] H. Pham. Some methods and applications of large deviations in finance and insurance. In *Paris-Princeton Lecture Notes in Mathematical Finance*. Springer, 2007.

[41] I. Psaras, R. G. Clegg, R. Landa, W. K. Chai, and G. Pavlou. Modelling and evaluation of CCN-caching trees. In *Proceedings of NETWORKING*, pages 78–91. Springer, 2011.

[42] L. Rizzo and L. Vicisano. Replacement policies for a proxy cache. *IEEE/ACM Transactions on Networking (ToN)*, 8(2):158–170, 2000.

[43] J. Roberts and N. Sbihi. Exploring the memory-bandwidth tradeoff in an information-centric network. In *Proceedings of 25th International Teletraffic Congress*, pages 1–9, 2013.

[44] E. J. Rosensweig, J. F. Kurose, and D. F. Towsley. Approximate models for general cache networks. In *Proceedings of IEEE INFOCOM*, 2010.

[45] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2 edition, 2003.

[46] R. K. Sitaraman, M. Kasbekar, W. Lichtenstein, and M. Jain. Overlay networks: An akamai perspective. In Pathan, Sitaraman, and Robinson, editors, *Advanced Content Delivery, Streaming, and Cloud Services*. John Wiley & Sons, 2014.

[47] H. A. Van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on scientific and Statistical Computing*, 13(2):631–644, 1992.