Unravelling the Impact of Temporal and Geographical Locality in Content Caching Systems

Stefano Traverso, Mohamed Ahmed, Michele Garetto,
Paolo Giaccone, Emilio Leonardi, Saverio Niccolini

***Abstract*—To assess the performance of caching systems, the definition of a proper process describing the content requests generated by users is required. Starting from the analysis of traces of YouTube video requests collected inside operational networks, we identify the characteristics of real traffic that need to be represented and those that instead can be safely neglected. Based on our observations, we introduce a simple, parsimonious traffic model, named Shot Noise Model (SNM), that allows us to capture temporal and geographical locality of content popularity. The SNM is sufficiently simple to be effectively employed in both analytical and scalable simulative studies of caching systems. We demonstrate this by analytically characterizing the performance of the LRU caching policy under the SNM, for both a single cache and a network of caches. With respect to the standard Independent Reference Model (IRM), some paradigmatic shifts, concerning the impact of various traffic characteristics on cache performance, clearly emerge from our results.**

I. Introduction

It is no surprise to find that the design and analysis of content caching systems continue to receive attention from both industry and academia. The big players in the market (Google, Akamai, Limelight, Level3, etc.), today preside over a multi-billion dollar business built on content delivery networks (CDNs), which employ massively distributed networks of caches to carry over half of Internet traffic, according to recent measurements [1]. To illustrate this reality, in 2010 Akamai listed its CDN to include over 60,000 servers in 1000 networks, spread over 70 countries [2]. The impressive growth of CDNs is essentially driven by the explosion of multimedia traffic. It is expected that video traffic alone will be around 70 percent of all consumer Internet traffic in 2017, and almost two-thirds of it will be delivered by CDNs [1].

The fundamental role played by caching systems goes beyond existing CDNs. Indeed, a radical change of communication paradigm may take place in the future Internet, from the traditional host-to-host communication model created in the 1970s, to a new host-to-content kind of interaction, in which the main networking functionalities are directly driven by object identifiers, rather than host addresses. In particular, Content-Centric-Networking proposals (CCN) [3] aim at redesigning the entire global network architecture with named data as the central element of the communication. This translates in practice into the need to redesign core routers by equipping them with fast, small caches, capable of processing requests at line speed. Nonetheless, to date, the design and evaluation of large-scale, interconnected systems of caches is still poorly understood. In the first place, it is unclear how to properly describe the traffic (in terms of the sequence of contents' requests) generated by the users, that is then processed by cache networks. In this regard, just resorting to trace-driven simulations to assess the performance of a cache architecture clearly has severe limitations, as we will elaborate in the next section. It is therefore highly desirable to have, first of all, a proper model for the arrival process of contents' requests at the caches. The main challenge here is to find a good compromise between: i) the fidelity of the model in describing the behavior of real traffic; and ii) its simplicity, which permits the development of analytical tools to predict the system performance. To fully address this problem, one needs to identify the traffic features playing the most crucial role for the resulting cache performance, and capture them into in a flexible, parsimonious, and analytically tractable manner. To the best of our knowledge, this problem has not yet received a satisfactory answer.

S. Traverso, P. Giaccone, E. Leonardi are with the Department of Electronic and Telecommunications, Politecnico di Torino, Italy: e-mail: {lastname}@tlc.polito.it. M. Ahmed, S. Niccolini are with the NEC Labs Europe, Heidelberg, Germany: e-mail: {name.lastname}@neclab.eu. Michele Garetto is with Università di Torino, Italy; e-mail: michele.garetto@unito.it.

A. The necessity of a traffic model

Although caching systems continue to attract interest in the networking community, there is still no common agreement on the traffic assumptions under which system design and performance evaluation should be carried out, especially in the context of pervasive CDN and CCN architectures. To obtain the best fidelity in evaluating a given system, one could simply follow the approach of performing trace-driven analysis [4], [5]. This approach, however, has several shortcomings. First, it does not enable us to identify the important factors that influence system performance and understand their role. Second, it does not permit us to explore "what if" scenarios, such as: how will my system perform if I increase/modify the catalogue of available contents? or if the users' population becomes much larger? Third, too often, we are constrained by the size and/or the availability of data sets, their diversity as well as legal and privacy concerns, which impacts the fidelity of the analysis.

To overcome these limitations, we can instead analyze caching systems using synthetic traces produced by a traffic model. The simplest, and still most widely adopted traffic model in the cache literature [6], [7], [8], [9] is the so-called Independent Reference Model (IRM) [10], according to which the sequence of content requests arriving at a cache is characterized by the following fundamental assumptions: i) there exists a fixed catalogue of N distinct contents, which does not change over time; ii) the probability a request is for a specific content is *constant* (i.e. the content popularity does not vary) and *independent* of all past requests. The IRM is commonly used in combination with a Zipf-like law of content popularity. In its simplest form, Zipf's law states that the probability to request the nth most popular content is proportional to $1/n^{\alpha}$, where the exponent α depends on the considered system (especially on the type of contents) [8], and plays a crucial role on the resulting cache performance.

By construction, the IRM completely ignores all temporal correlations in the sequence of requests. In particular, it does not take into account a key feature of real traffic, usually referred to as *temporal locality*, which occurs when requests for a given content densify over short periods of time (with respect to the trace duration). The important role played by temporal locality, especially its beneficial effect on cache performance, is well known in the context of computer memory architecture [10] and web traffic [11]. Indeed, several extensions of IRM have been already proposed to incorporate temporal correlations in the request process [10], [11], [12], [13], i.e., the fact that, if a content is requested at a given time instant, then it is more likely that the same content will be requested again in the near future. Existing models, however, have been primarily thought for web traffic, and they share the following two assumptions: i) the content catalogue is fixed; ii) the request process for each content is stationary (i.e., either a renewal process or a semi-Markov-modulated Poisson process or a self-similar process). As we will see, these assumptions are not appropriate to capture the kind of temporal locality usually encountered in Video-on-Demand traffic, because they do not easily capture intrinsically non-stationary macroscopic effects related to content popularity dynamics. Moreover some of the previously proposed models are too complex to allow an analytical study of caching systems.

At the other extreme, some recent studies have proposed rather sophisticated models describing the evolution of contents popularity at the macroscopic level. These show that the aggregate download process of popular on-line contents is highly non-stationary and exhibits a complex correlation structure resulting from social cascades and other viral phenomena [14], [15]. Fairly complex stochastic models, i.e. based on Hawkes processes [14] or autoregressive (ARIMA) models [15], have been recently proposed to accurately describe the large-scale content popularity evolution. However, it is unclear how these models can be used to generate a synthetic sequence of content requests arriving at a specific cache, which aggregates traffic from a limited number of users.

Furthermore, with respect to distributed systems of caches, the *geographical locality* of user requests has been little investigated in the literature and largely ignored by existing analytical models. Large-scale, pervasive systems of caches typically serve heterogeneous communities of users having different interests, and therefore the probability of a request for a given content can vary significantly from region to region. The studies that have appeared [16], [17], [18], show that geographical locality is (as expected) hardly observable within culturally homogeneous regions, and becomes evident in large systems serving different linguistic/cultural communities. Our results (see Sec. II-B) show that geographical locality can be observed to some extent even in limited geographical regions because users associated to different caches vary in their social/ethnic/linguistic composition.

To the best of our knowledge, no traffic models have been proposed so far to incorporate various degrees of geographical locality in the content request processes arriving at different caches. Furthermore, the impact of geographical locality on cache performance, especially in distributed systems of interconnected caches, is still poorly understood [19], [20].

B. Paper contributions

Because of its dominant and growing role in the Internet, in this paper we focus on video traffic, specifically, Video-on-Demand (VoD). Nonetheless, our methodology and results have broader applicability, and may also be of interest to other kinds of contents. We are especially interested in pervasive CDN or CCN architectures, comprising several caches (which can be small relative to the content catalogue size) serving localized communities of users. However, in this work, we do not consider the case in which users are spread over many different time zones.

Our first main contribution is a new traffic model describing the contents' request processes originated by the users, to be used in input to the edge caches of the system. In pursuing this goal, we aimed at filling the existing gap between simple, stationary traffic generators developed for traditional caching systems (computer architecture, web traffic) [10], [11] and the complex stochastic models describing macroscopic worldwide content popularity dynamics [14], [15]. Our proposed traffic model meets the following requirements: (1) be general and flexible; (2) provide a native explanation for the temporal and geographical locality in the request process; (3) explicitly represent content popularity dynamics; (4) capture the phenomena having major impact on cache performance, while neglecting those with no or limited impact; (5) be as simple as possible while maintaining accuracy; and (6) permit developing analytical models of popular caching policies.

Our second main contribution is an accurate analysis of the LRU (Least Recently Used) caching policy under the newly proposed traffic model, that provides fundamental insights on the impact of several traffic characteristics on cache performance, which have not been documented before.

In more detail, we provide the following contributions, listed in the sequence in which they are derived in the paper: **(C1)** We analyze the temporal and geographical locality of real Video-on-Demand traffic collected in six different locations (Sec. II). **(C2)** We show that the standard IRM approach to modeling users' contents requests leads to significant errors when estimating the cache size needed to achieve a given hit ratio, especially when cache sizes are small relative to the catalogue size (Sec. II-A). **(C3)** We propose and validate (using our traces) the Shot Noise Model (SNM), a more accurate and flexible model alternative to the IRM (Sec. III). **(C4)** We show that, in contrast to common expectation, daily variations in the aggregate request rate, as well as the detailed shape of the popularity profile of individual contents, have negligible impact on cache performance, advocating the idea of a parsimonious traffic model (in terms of the number of parameters) (Sec. III-B). **(C5)** We explain how the SNM can be extended to the case of a cache network, validating some simplifying assumptions that we propose to incorporate geographical locality in the model (Sec. III-C and III-D). **(C6)** We *analytically* characterize the performance of the LRU caching policy under the SNM, providing enlightening closed-form expressions for the large and small cache regimes (Sec. IV). **(C7)** Using numerical and simulative analysis, we explore the effect of a wider range of model parameterizations than the one available from the traces (Sec. V), gaining deeper insights into the impact of several traffic parameters, which in some cases depart from the general understanding gained using

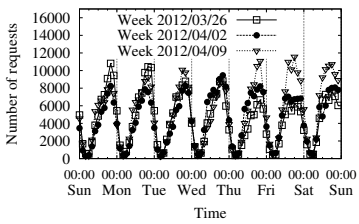


Fig. 1: Evolution of the volume of requests over three weeks for *Trace 3*.

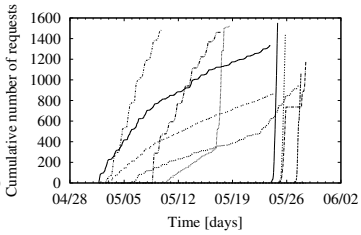


Fig. 2: Cumulative number of requests over time for a subset of videos in *Trace 2*.

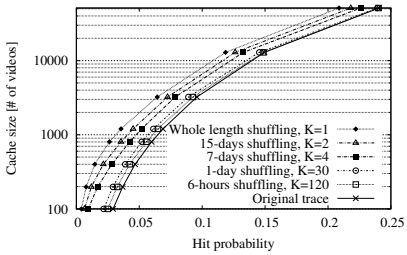


Fig. 3: The cache size required to achieve a desired hit probability, when an LRU cache is fed by the requests contained in *Trace 1*, subject to different degrees of trace reshuffling.

the IRM. (Sec. V-A). **(C8)** At last, we show how our traffic model and analysis can be effectively used for system design and optimization, investigating some additional examples of traffic mixes (Sec. V-B and V-C).

II. A Dive into Real Traffic

We employed Tstat[1] an open-source traffic monitoring tool to analyze TCP/IP packets sent/received by actual end-users, captured at monitored vantage points. Probes were installed in five PoPs located at different cities of two different countries, Italy and Poland. Table I provides details about our vantage points inside the network. Probes *Home 1*, *Home 2*, *Home 3* and *Home 4* are located in three cities in Italy, and monitor the traffic of about 65,000 residential customers of a large ISP offering Internet access by ADSL and FTTH technologies. Similarly, *Home 5*, located in Poland, monitors the network activity of approximately 5,000 residential customers. Finally, probe *Campus 1* was deployed within the network backbone of Politecnico di Torino in Italy, which provides Internet access to about 15,000 students mainly through Wi-Fi access.

Table II details the ten traces employed in our study. Measurements were performed on both incoming and outgoing traffic over two different periods (March-May 2012 and February-April 2013) and together cover approximately 6 months. In total, we observed the activity of about 85,000 end-users accessing the Internet, and identified the TCP flows corresponding to YouTube video requests and downloads. In total, we recorded more than 20 million transactions.

A. Temporal locality

From analyzing the traces, we observe two main factors responsible for the temporal locality in the sequence of requests made by users. First, the *aggregate* arrival rate of requests follows the expected diurnal variation, as shown in Fig. 1. Second, the arrival rate of requests for a given content can be highly non-stationary, being often concentrated in intervals much shorter than the total trace duration. Furthermore, as illustrated in Fig. 2, contents display a wide range of popularity evolution patterns [21]. Although these facts are well known and have been already examined before, their impact on cache performance must be carefully evaluated.

With regard to diurnal variations, we observe that, contrary to what is sometimes believed [21], accounting for this variation has no impact on the main performance metrics for caches such as the hit probability (see Sec. IV-A5). To intuitively understand why this is the case, consider that the hit probability of almost all proposed caching policies depends only on the sequence of content identifiers arriving at the cache, and not on the time-stamps associated with the requests. Therefore, if we arbitrarily squeeze or stretch (over time) the aggregated sequence of content requests arriving at a cache, we obtain the same hit probability (this holds for all caching policies which do not explicitly use the information about the request arrival time). For this reason, in our synthetic traffic model we will ignore the diurnal rate variations.

On the other hand, the non-stationarity of the popularity of individual contents has a significant impact on the performance of a cache. Accounting for this property is a difficult task due to the complexity and heterogeneity of content popularity dynamics. For example, the popularity of some videos vanishes after only a few days, while others continue to attract requests for significantly long periods of time (months or even years). Besides the life-span, clearly also the number of requests attracted by the videos can be very diverse.

To better understand the impact on cache performance of the temporal locality present in our traces, we carried out the following experiment: we fed an LRU cache (initially empty, and transient effects have been filtered out in the evaluation) with the sequence of requests contained in *Trace 1* (similar results were obtained using the other traces) and derived the cache size necessary to achieve a given cache hit probability. Results are reported in Fig. 3 by the curve labeled "Original trace". Then, we partitioned *Trace 1* into K slices, each containing an equal number of requests, and we randomly permuted the requests within each slice. Such artificially shuffled traces (one for each K) are then fed again into an LRU cache, deriving again the required cache size to achieve a given hit probability. Different values of K correspond to washing out the temporal locality at a time scale equal to the corresponding slice duration (for clarity, the approximate slice duration is also reported in the legend of Fig. 3, for each considered value of K). Note that the original trace can

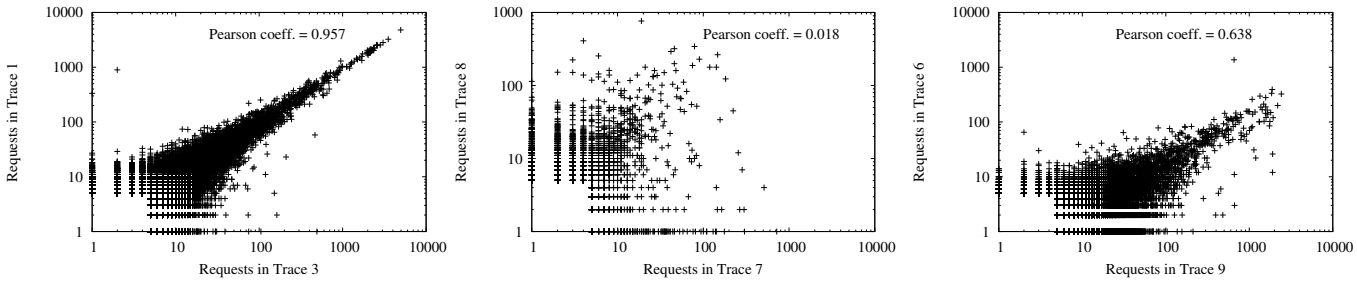
[1] http://www.tstat.polito.it

Probe	Type	IPs	Probe	Type	IPs
Home 1	ADSL	16172	*Home 4*	ADSL	2543
Home 2	ADSL/FTTH	17242	*Home 5*	ADSL	5080
Home 3	ADSL/FTTH	31124	*Campus 1*	LAN/Wi-Fi	15000

TABLE I: Probe characteristics.

Probe	Trace	Period	Length	Requests	Videos
Home 1	*Trace 1*	20/03/12-25/04/12	35 days	1.7M	0.93M
	Trace 2	30/04/12-28/05/12	27 days	1.8M	0.95M
	Trace 6	18/03/13-24/04/13	36 days	1.6M	0.86M
Home 2	*Trace 3*	20/03/12-30/04/12	40 days	2.4M	1.24M
	Trace 7	12/03/13-24/04/13	42 days	2.4M	1.22M
Home 3	*Trace 4*	20/03/12-25/04/12	35 days	3.8M	1.76M
	Trace 10	18/03/13-24/04/13	36 days	3.7M	1.69M
Home 4	*Trace 5*	15/02/13-23/04/13	67 days	0.4M	0.25M
Home 5	*Trace 8*	28/02/13-21/03/13	21 days	0.6M	0.28M
Campus 1	*Trace 9*	7/03/12-13/05/12	67 days	0.55M	0.35M

TABLE II: Measurement traces.



(a) *Trace 1* and *Trace 3* (same country). (b) *Trace 7* and *Trace 8* (different countries). (c) *Trace 6* and *Trace 9* (same country).

Fig. 4: Scatter-plot of the number of content requests in trace pairs. Each point corresponds to a specific content.

be considered as a limit case of very large K (equal to the number of requests in the trace), whereas $K = 1$ corresponds to the case in which we randomly permute the entire trace, destroying all temporal correlations within the whole trace duration (about one month).

As expected, we find that temporal locality plays a significant role on cache performance – the cache size needed to achieve a desired hit probability increases considerably between the original trace and the extreme case with $K = 1$. Now, suppose that we were to completely ignore the effects of temporal locality, by adopting a naive IRM approach in which we just compute from the trace the empirical popularity distribution for the contents requested in the trace, and use such empirical distribution as the popularity law of the IRM. The hit probability achieved for a given cache size, according to the above IRM model, would be essentially equivalent to the one derived in our experiment in the case $K = 1$. Indeed, the complete trace shuffling leads to an i.i.d. sequence of requests following the long-term empirical probability distribution of the trace, just as in the considered IRM model. In conclusion, the adoption of a naive IRM approach for VoD contents leads to considerably erroneous (pessimistic) estimates of cache performance, especially when caches are small.

We also observe that, as the slice duration approaches the order of a few hours, the required cache size becomes very close to the one resulting from the original trace. This means that: i) the evolution of content popularity over timescales of few days/weeks is important for the resulting cache performance; ii) we can, instead, ignore short timescale effects (i.e., correlations taking place over timescales of few hours or less). The practical consequence of this is that we do not need to take into account complex fine-grained correlations in the arrival process of requests (in particular at the level of contents' inter-request times). Actually, given that short time-scale correlations (up to a few hours) are not important to predict cache performance for VoD contents, we can well adopt (locally) a Poisson approximation for the arrival process of requests, which enables us to build simple analytical models of cache behavior, such as those developed in Sec. IV.

We remark that our findings are in sharp contrast with what has been observed in the case of web traffic, whereby the impact of short time-scale correlations greatly outweighs that due to long-term correlations [22]. This observation signifies the different nature of VoD with respect to web browsing, motivating the development of new models specifically tailored to this kind of traffic.

We emphasize that the IRM approach could, in principle, still be adopted to effectively predict the hit probability in the scenario considered in Fig. 3. This would require us to estimate from the trace a proper content catalogue size (which is a non-trivial task in its own) and to properly set the contents popularity law of IRM so as to match the short-term (say, over a few hours) distribution of content popularity observed in the trace. Estimating a short-term popularity distribution from a trace, however, is a challenging task (as recognized in [9]), being any measurement collected over a short period necessarily affected by a large amount of noise. Moreover, the proper timescale at which the IRM could be effectively employed is hard to predict, as it depends on many parameters (cache size, caching policy, arrival rate of requests, etc.), forcing us to compute a different short-term popularity distribution for each considered scenario. On the contrary, the parameters of our traffic model can be derived, once for all, from long-term measurements, and our analysis of cache performance does not require to explicitly compute any short-term popularity law.

B. *Geographical locality*

Geographical diversity in the contents' popularity is expected to play a significant role in a large-scale systems of interconnected caches, in which edge caches receive the requests of subsets of users geographically localized in the same region, and thus likely to share similar interests. Thanks to the significant time-overlap between some of the traces belonging to our data set (see Table II), we were able to assess, to some extent, the geographical locality of VoD traffic. In particular, we counted the number of requests received by each video, during the largest common time interval between two given traces of our data set. Fig.s 4a-4c show the resulting number of requests in one trace vs. the one in the other trace, for three significant cases. Note that a perfect linear relation here would imply exactly the same relative popularity of contents in the two different traces, corresponding to a Pearson correlation coefficient equal to 1. In Fig. 4a we consider two traces collected at probes belonging to the same residential ISP in one country (Italy), i.e., serving highly homogeneous users in terms of language and culture. As expected, contents which are very popular in one trace are also very popular in the other trace, as confirmed by the Pearson coefficient very close to 1. In contrast, Fig. 4b considers two traces collected at probes located in different countries (Italy and Poland), whose native languages are different. In this case, we observed a very low correlation (Pearson coefficient close to 0), suggesting that the cultural background of users plays a crucial role in the

diversity of the contents request process[2].

Although country borders may represent a good baseline to identify clusters of homogeneous users, we observed that even within the same country there can be significant heterogeneity in terms of users' interests. In Fig. 4c we compare a trace collected in a residential network (PoP *Home 1*) with the trace from a university campus network (*Campus 1*), on a time-overlap of about one month. Even though both traces are collected in the same city, a significantly higher fraction of points lie far from the diagonal, with respect to the scenario in Fig. 4a, especially in the case of popular videos. This is confirmed by the correlation coefficient, here equal of 0.638.

We conclude that accounting for geographical locality in a traffic model can be a difficult task due to cultural/social effects. A deep assessment of the amount of geographical diversity in the network, by means of large measurement campaigns, would be required to build accurate synthetic traffic models. Unfortunately, due to the limited available data set, we were able to investigate the impact of geographical locality on cache performance only based on synthetic traffic traces, as discussed later in Sec. V-C.

III. Shot noise traffic model

Guided by the insights gained from our traces (Sec. II), we propose a new traffic model, aimed at striking a good compromise between simplicity, flexibility and accuracy. We first consider the single cache case and then move on to the case of a network of caches.

A. Basic model for a single cache

The rationale of our traffic model is to capture the physical origin of the temporal locality observed in the traces. Our solution is to represent the overall request process as the superposition of many independent processes, each referring to an individual content. As such, the arrival process of a given content m is specified by three physical parameters (τ_m, V_m, $\lambda_m(t)$): τ_m represents the time instant at which the content enters the system (i.e., it becomes available to the users); V_m denotes the average number of requests generated by the content; $\lambda_m(t)$ is the "popularity profile", describing how the request rate for content m evolves over time. In general, function $\lambda_m(t)$ satisfies the following conditions: (positiveness) $\lambda_m(t) \geq 0$, $\forall t$; (causality) $\lambda_m(t) = 0$, $\forall t < 0$; (normalization) $\int_0^\infty \lambda_m(t)\,\mathrm{d}t = 1$. We define the *average life-span* of content m, L_m, as follows:

$$L_m = \frac{1}{\int_0^\infty \lambda_m^2(t)dt} \tag{1}$$

It will become clear later, while analyzing the performance of LRU in Sec. IV, why it is convenient to define the life-span of a content using the formula above. For now, to get an initial understanding of the definition, consider a content with a uniform popularity profile $\lambda_m(t) = 1/\delta$ for $t \in [0, \delta]$. By computing (1), we obtain $L_m = \delta$, which is the intuitive value of life-span of such content.

Given the above parameters, our model assumes that the request process for content m is described by a time-inhomogeneous Poisson process whose instantaneous rate at time t is given by $V_m \cdot \lambda_m(t - \tau_m)$.

[2]We verified that the few YouTube videos attracting a large volume of requests in both traces considered in Fig. 4b are related to famous international hits.

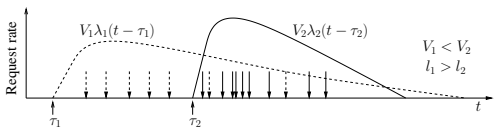


Fig. 5: Example of requests (denoted by arrows) generated by two contents with different catalogue insertion time (τ_1, τ_2), average number of requests (V_1, V_2) and profiles ($\lambda_1(t)$,$\lambda_2(t)$).

For the sake of simplicity, we assume that new contents become available in the system according to a homogeneous Poisson process of rate γ, i.e., time instants $\{\tau_m\}_m$ form a standard Poisson process. We refer to this model as Shot Noise Model (SNM), since the overall process of requests arrival is known as a Poisson shot-noise process [23]. Fig. 5 illustrates an example of the request pattern generated by the superposition of two "shots" corresponding to two contents having quite different parameters. We emphasize that the above Poisson assumptions, on the (instantaneous) generation process of requests for each content, and on the arrival process of new contents, are essentially introduced for the sake of analytical tractability. However, they are very well justified by the experience gained from our traces, which show that it is not really important to capture complex arrival dynamics at short time-scales (see discussion about the results in Fig. 3).

For a given content, the SNM requires us to specify its entire popularity profile in the form of the function $\lambda_m(t)$, which, given the difficulty in estimating popularity profiles from a trace, could be considered as a limitation. However, we have found that *it is not necessary to precisely identify the shape of* $\lambda_m(t)$. In fact, a simple first-order approximation, according to which we just specify the average content life-span L_m, is enough to obtain accurate predictions of cache performance. In other words, we can arbitrarily choose any reasonable function $\lambda_m(t)$ with an assigned life-span L_m, and obtain almost the same results in terms of cache performance (see the later discussion on Fig. 6). Finally, content heterogeneity is taken into account by associating to every content, its life-span L_m, jointly with the (typically correlated) average number of requests V_m. This means that, upon arrival of each new content m, we randomly choose (independently for each content) the pair of parameters (V_m, L_m) from a given assigned joint distribution.

B. Validation of basic SNM

To show how our traffic model can accurately capture the temporal locality observed in real data, and its impact on cache performance, we introduce a simple procedure to fit its parameters from a trace.

For each given content in the trace first we compute the total number of observed requests $\hat{V}_m$, and then we estimate the content life-span $\hat{L}_m$. The interested reader can found the details about how to estimate $\hat{L}_m$ from the traces in the preliminary version of this paper [24]. Contents are then partitioned into 6 classes $(0, \ldots, 5)$, on the basis of previous values $\hat{V}_m$ and $\hat{L}_m$. Class 0 comprises all the contents with small number of requests ($\hat{V}_m < 10$), for which we cannot derive a reliable estimate of their life-span. Contents in classes 1 to 5 contain contents with $\hat{V}_m \geq 10$ which, as reported in Table III, are partitioned according to $\hat{L}_m$ (measured in days). For each class, Table III reports, for four different traces,

Class	Classification rule	Trace	%Reqs	%Videos	$\mathbb{E}[\hat{L}_m]$	$\mathbb{E}[\hat{V}_m]$
Class 0	$\hat{V}_m < 10$	*Trace 1*	74.60	98.588	-	1.41
		Trace 2	72.64	98.401	-	1.42
		Trace 3	72.53	98.210	-	1.44
		Trace 4	67.30	97.778	-	1.49
Class 1	$\hat{V}_m \geq 10$ $\hat{L}_m \leq 2$	*Trace 1*	2.34	0.044	1.14	86.4
		Trace 2	2.71	0.083	1.09	76.2
		Trace 3	2.60	0.067	1.04	76.0
		Trace 4	2.81	0.077	1.06	74.0
Class 2	$\hat{V}_m \geq 10$ $2 < \hat{L}_m \leq 5$	*Trace 1*	1.72	0.069	3.36	41.9
		Trace 2	3.43	0.125	3.34	50.7
		Trace 3	1.77	0.082	3.32	43.3
		Trace 4	2.01	0.093	3.41	48.0
Class 3	$\hat{V}_m \geq 10$ $5 < \hat{L}_m \leq 8$	*Trace 1*	1.49	0.041	6.40	59.5
		Trace 2	1.84	0.070	6.31	44.9
		Trace 3	1.66	0.052	6.42	63.3
		Trace 4	1.64	0.062	6.45	60.3
Class 4	$\hat{V}_m \geq 10$ $8 < \hat{L}_m \leq 13$	*Trace 1*	1.39	0.062	10.53	36.9
		Trace 2	2.96	0.128	10.86	39.6
		Trace 3	1.33	0.066	10.62	39.5
		Trace 4	1.75	0.103	10.65	37.8
Class 5	$\hat{V}_m \geq 10$ $\hat{L}_m > 13$	*Trace 1*	18.46	1.196	24.61	25.7
		Trace 2	16.41	1.193	19.29	25.3
		Trace 3	20.11	1.523	28.19	25.8
		Trace 4	24.49	1.887	24.59	28.1

TABLE III: Model parameters for each content class. $\hat{L}_m$ is evaluated in days. The numbers of requests and videos for classes 0–5 have been normalized separately for each trace.

Profile	$\lambda(t)$	L
Uniform	$1/\delta$ for $t \in [0, \delta]$	δ
Exponential	$(1/\delta)e^{-t/\delta}$ for $t \geq 0$	2δ
Power law ($\zeta > 1$)	$\frac{\zeta - 1}{\delta}\left(\frac{t}{\delta} + 1\right)^{-\zeta}$ for $t \geq 0$	$\frac{\delta(2\zeta - 1)}{(\zeta - 1)^2}$

TABLE IV: Popularity profile and corresponding life-span L.

the percentage of total requests attracted by the class, the percentage of videos belonging to it, and the average values $E[\hat{L}_m]$ and $E[\hat{V}_m]$. for the videos in the class.

From Table III, we observe that: (i) The values related to each class are quite similar (with differences of at most 20%) across the considered traces. This is significant, because it suggests that our broad classification captures some invariant properties of the considered VoD traffic. (ii) Contents in Class 1 (having $\hat{L}_m < 2$ days) represent roughly 0.07% of the total number of contents (4% of contents in Classes 1-5), but account for approximately 2.5% of all requests (10% of requests originated by contents in Classes 1-5). These contents exhibit the larger degree of temporal locality, and we will see that their impact on cache performance is crucial, despite the fact that they represent a rather small fraction of the traffic. (iii) Contents in Class 5 have a life-span comparable with the trace length, therefore their measured value $\hat{L}_m$ is expected to be strongly affected (i.e., underestimated) by border effects due to the finiteness of the trace. We chose not to attempt to characterize the temporal locality of contents belonging to either Class 0 (too few requests) and Class 5 (unreliable estimate of life-span), by using the SNM. We therefore treat these contents as if their popularity was stationary (like in the IRM), and generate their requests uniformly in the considered time horizon. We emphasize that, with this choice, we miss the opportunity to capture the temporal locality of a large fraction of contents. On the other hand, by so doing we obtain a conservative prediction of cache performance.

As such, only the requests for contents falling in Classes 1 to 4 are generated according to the SNM model, assuming a common "shape" $\lambda(t)$ for the popularity profile, chosen from the profiles listed in Table IV. For each content class, the shape parameter δ is chosen to match the average life-span $\mathbb{E}[\hat{L}_m]$. Moreover, for each class modeled by the SNM, the distribution of request volumes was matched to the corresponding empirical distribution observed in the trace.

We generated a synthetic request trace using the parameters estimated as above, and fed it to a cache implementing the LRU policy. Fig. 6 reports the cache size required to achieve a desired hit probability, using *Trace 4* (similar results were obtained with the other traces). For comparison, we report also the results obtained with the original trace and those obtained by its completely shuffled version representing the "naive IRM" approach. We observe that the results obtained by applying the fitted SNM (using either uniform, exponential or power-law shape with $\zeta = 3$) are very close to those obtained with the original unmodified trace, and that the shape chosen for the popularity profile has little impact on the results.

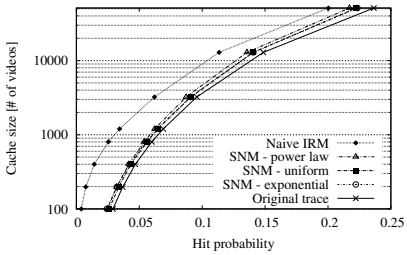


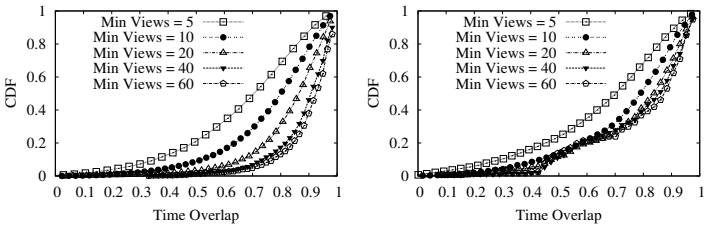
Fig. 6: Cache size vs hit probability under LRU, for *Trace 4*.

In summary, Fig. 6 shows that our SNM provides an accurate prediction of cache performance, despite the heavy simplifications adopted in the parameters' identification. We expect that even more accurate predictions could be achieved by improving the fitting procedure, or by using much longer traces (if available).

C. Extension of SNM to cache networks

We now extend the basic SNM introduced in Sec. III-A to handle the case of multiple interconnected caches, in which edge caches receive the requests generated by subsets of users possibly having quite different interests from one ingress point to another (geographical locality). The extension of the basic model can be, in principle, carried out in its most general form by associating to every content m and ingress point $i \in I$, a tuple $(\tau_{m,i}, V_{m,i}, \lambda_{m,i})$, so that, at ingress point i, requests for content m arrive according to an inhomogeneous Poisson process, whose instantaneous rate at time t is given by $V_{m,i} \cdot \lambda_{m,i}(t - \tau_{m,i})$. While this approach is fairly general, it is not very practical, because the total number of parameters to be specified may become very large. An alternative is to make the following simplifications. First, we assume that the instants at which content m starts to be available in the system at the various ingress points ($\tau_{m,i}$) are equal, i.e., $\tau_{m,i} = \tau_m$. This is well justified since in most systems contents are enabled to be globally available to all users at the same time. Second, the popularity of a given content m is assumed to follow the same profile $\lambda(t)$ at every ingress point. Together with the previous consideration regarding the negligible impact of the particular shape of the popularity profile, we represent each content m by: i) its (global) starting point of availability τ_m; ii) its (global) life-span L_m; iii) a set $\{V_{m,i}\}_i$ of (local) parameters denoting the volumes of requests attracted by content m at different ingress points.

Our simplifications are realistic for cache networks covering limited geographical areas. We remark that, when considering content distribution systems spanning large areas, including very different time zones (i.e. at world-wide scale), temporal profiles (especially for contents having small life-span) should



(a) *Trace 6* and *Trace 7* (same country).

(b) *Trace 3* and *Trace 8* (different countries).

Fig. 7: Time overlap for different pairs of traces.

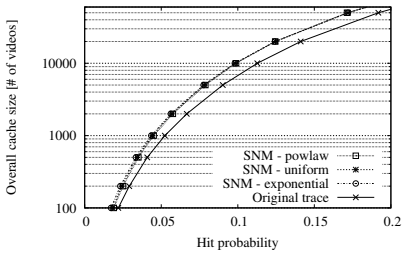


Fig. 8: Overall cache size (60% in the root, 20% in each leaf) vs hit probability, in a cache network fed either by real traces or by synchronized synthetic traffic.

not be considered synchronized, but properly modified to take jointly into account geographical locality and diurnal patterns.

Finally, to specify the volumes of requests generated by contents at the various ingress points ($V_{m,i}$), we adopt the following approach: for each content m, we assign a global volume V_m, denoting the total number of requests generated in the whole system. Then, we specify $V_{m,i}$ as $V_{m,i} = V_m \cdot p_{i,m}$, where $p_{i,m}$ represents the fraction of requests for content m arriving at ingress point i. By construction $p_{i,m} \geq 0$ and $\sum_i p_{i,m} = 1$. This approach allows a large degree of flexibility in describing the geographical locality of contents. For example, we can obtain the case in which geographical locality is negligible, by setting $p_{i,m} = p_i$, for every m; here p_i represents the "relative mass" of requests arriving at ingress point i (i.e., the fraction of all requests generated by the corresponding set of users). At the other extreme, we can represent the case of complete geographical locality by setting $p_{i,m} \in \{0, 1\}$ (i.e., setting each content to be requested at only one particular ingress point).

D. Validation of SNM for cache networks

To assess the validity of our simplifying assumptions, according to which the popularity evolution of each content is "synchronized" across different ingress points (i.e., $\tau_{m,i} = \tau_m$ and $\lambda_{m,i} = \lambda_m$, $\forall i$), we evaluated the degree of temporal overlap between the sequence of requests received by a content in different traces of our data set. In particular, given a pair of traces, we computed the following metric for each content m which appears in both traces:

$$\text{time-overlap-fraction}(m) = \frac{|\text{intersection of life-spans}|}{|\text{union of life-spans}|}$$

Note that, by definition, the above metric takes values in $[0, 1]$, where 0 represents no overlap and 1 is produced by identical and perfectly overlapped life-spans. For example, the case of two equal life-spans, shifted in time by 20%, leads to time-overlap-fraction $= 0.8/1.2 = 0.67$.

Fig.s 7a and 7b, respectively report the CDFs of the time-overlap-fraction for *Trace 6* and *Trace 7* (collected from residential networks in the same country) and for *Trace 3* and *Trace 8* (collected from residential networks in different countries). To get additional insight, we obtained a separate CDF for the contents which receive at least a certain minimum number of requests in both traces (this is the parameter "Min Views" reported in the figures, ranging from 5 to 60). We observe that the degree of synchronization increases with the content popularity. This can be in part due to the fact that the life-span interval of a content resulting from a trace is affected randomly by border effects, which smooth out as the request volume increases. Indeed, even when the popularity evolution of a content is perfectly synchronized across different ingress points, i.e. $\tau_{m,i} = \tau_m$ and $\lambda_{m,i} = \lambda_m$, the overlap observed in the actual sequence of requests can be small for contents with few requests. Nevertheless, we observe quite a strong degree of synchronizing in both Fig.s 7a and 7b. For example, only about 25% of the contents receiving more than just 10 requests show a time-overlap-fraction smaller than 0.67.

To show how the life-span of a content measured in different traces can vary in size, Table V reports, for the same two pairs of traces considered before, the fraction of contents that, given an initial classification in one trace (the row), were classified into a given class (the column) in the second trace (we refer to the definition of classes in Table III). By construction, each row in the table sums to one. We observe that the majority of contents are classified into the same class in both traces. Moreover, the fraction of contents for which the class index differs by more than one is negligible. Note that contents whose class index differs exactly by one (i.e., the cells in the first diagonal above or below the main diagonal) should be taken with care, because their life-spans might be close to the threshold value separating two neighboring classes, which can easily lead to a misclassification.

Having observed that the popularity evolution of contents appears to be well synchronized across the traces (both in terms of overlap and size of the corresponding life-span intervals), we performed one more experiment to validate our modeling approach, to check how the non-perfect synchronization existing in real traces can affect the resulting cache performance. In particular, we considered a simple cache network composed by one root and two leaves. We assume that the root takes 60% of the overall cache size and each leaf takes 20% of it. Then we fed the left leaf by *Trace 1* and the right leaf by *Trace 3*. These traces where chosen because they contain similar request volumes and have a large temporal intersection. In the case of a miss, requests are forwarded to the root. Contents are replicated on all caches traversed by a

Final class		Class 1	Class 2	Class 3	Class 4	Class 5
Initial class		*Trace 7*				
Trace 6	Class 1	*0.69*	0.23	0.04	0.02	0.02
	Class 2	0.11	*0.64*	0.16	0.05	0.04
	Class 3	0.03	0.21	*0.44*	0.16	0.16
	Class 4	0.01	0.07	0.18	*0.35*	0.39
	Class 5	0.00	0.00	0.01	0.02	*0.97*
		Trace 8				
Trace 3	Class 1	*0.86*	0.06	0.04	0.00	0.01
	Class 2	0.30	*0.56*	0.08	0.03	0.03
	Class 3	0.10	0.27	*0.39*	0.13	0.11
	Class 4	0.02	0.12	0.26	*0.44*	0.16
	Class 5	0.00	0.01	0.02	0.03	*0.94*

TABLE V: Cross-classification of contents for two different pairs of traces.

request.

To obtain an easily tractable traffic model for this scenario, relying on the synchronization assumption, we proceeded as follows: we merged the two traces, and derived a unique SNM from the combined trace, using the same fitting procedure described in Sec. III-B. The requests in the synthetic trace produced by the SNM are then randomly distributed between the two leaves, in proportion to the number of requests in the original traces. Note that, by so doing, contents are assumed to be perfectly synchronized between the two ingress points.

Fig. 8 shows the overall cache size needed to obtain a given hit probability. The curve labeled "Original trace" refers to the case in which the network is fed by real traffic traces, whereas the other curves refer to synthetic traffic traces produced by the fitted SNM, using different shapes for the popularity profile. We observe that, despite all approximations and simplifying assumptions, our traffic model provides good predictions of cache performance, being the required cache size overestimated by a factor always lower than 2.

IV. Analysis of LRU under SNM

The Shot Noise Model introduced and validated in Sec. II is simple enough to permit developing accurate analytical models of classic caching policies. In particular, in this section we show how the Least Recently Used (LRU) caching policy can be analyzed under the traffic produced by the SNM, by extending a technique known as Che's approximation [25]. Note that, to ease the readability, the extension of our analysis to cache networks was moved to Appendix B.

A. The single cache case

Consider a cache capable of storing C distinct contents. Let $T_C(m)$ be the time needed for C distinct contents, not including m, to be requested by users. $T_C(m)$ therefore gives the *cache eviction time* for content m, i.e., the time since the last request for content m, after which content m is evicted from the cache. Of course $T_C(m)$ is a stochastic variable whose distribution typically depends on the considered content m; Che's approximation is based on the simplifying assumption that the cache eviction time ($T_C(m)$) is *deterministic* and *independent* of the considered content (m). This assumption has recently been given a theoretical justification in [9], where it was shown that, under IRM with a Zipf-like (static) popularity distribution, the coefficient of variation of $T_C(m)$ tends to vanish as the cache size grows. Furthermore, the dependence of the eviction time on m becomes negligible when the content catalogue is sufficiently large. Moreover, in [9] authors discover that Che's approximation is also surprisingly accurate in critical cases (small catalogue, very skewed popularity law). The arguments used in [9] are easily extended to our non-stationary traffic model when the product $\gamma \cdot \mathbb{E}[L_m]$ (the average number of concurrently "active" contents) and C are sufficiently large.

We start our analysis by considering the single-class case, in which the popularity profile ($\lambda(t)$) is the same for all contents, being characterized by the average content life-span L. The request volumes (V_m) are assumed to be i.i.d. random variables, distributed as V, with $\mathbb{E}[V] < \infty$. Finally, we define $\phi_V(x) = \mathbb{E}[e^{xV}]$ to be the moment generating function of V and $\phi'_V(x) = \mathbb{E}[Ve^{xV}]$ its first derivative, such that $\phi'_V(x) \geq 0$ for any x. Heterogeneity of the contents' popularity is handled by a multi-class extension that will be described later in Sec. IV-A4.

1) Main result for the single-class model: In the case of a single class of contents, the application of Che's approximation to analyze LRU performance under the SNM leads to the following fundamental result:

Theorem 1: Consider a cache of size C implementing LRU policy, operating under a SNM request arrival process with total stochastic intensity:

$$\Lambda(t) = \sum_{m:\tau_m<t} V_m \lambda(t-\tau_m)$$

representing τ_m points of an homogeneous Poisson process with intensity γ and V_m i.i.d. random variables. Under the Che's approximation, the hit probability is given by:

$$p_{\text{hit}} = 1 - \int_0^\infty \lambda(\tau) \frac{\phi'_V\left(-\int_0^{T_C} \lambda(\tau-\theta)d\theta\right)}{\mathbb{E}[V]} d\tau \quad (2)$$

where T_C is the only solution to equation:

$$C = \gamma \int_0^\infty 1 - \phi_V \left(-\int_0^{T_C} \lambda(\tau-\theta)d\theta\right) d\tau \quad (3)$$

Proof: The proof is reported in Appendix A. ∎

2) Small-cache regime: For small cache sizes, it is possible to derive a closed-form approximation of (2) and (3) as follows:

Corollary 1: If $\mathbb{E}[V^2] < \infty$, for small cache sizes the hit probability is approximated by:

$$p_{\text{hit}} \approx \frac{T_C}{L} \frac{\mathbb{E}[V^2]}{\mathbb{E}[V]} \quad (4)$$

where T_C derives from equation:

$$C = \gamma \mathbb{E}[V] T_C \quad (5)$$

Proof: The expression in (4) is obtained from (2) by approximating $\int_0^{T_C} \lambda(\tau-\theta)d\theta$ with $\lambda(\tau)T_C \ll 1$, and by locally approximating $\phi'_V(x)$, in a right neighborhood of $x = 0$, with its linear Taylor expansion: $\phi'_V(x) = \mathbb{E}[Ve^{-xV}] \approx \mathbb{E}[V] - x\mathbb{E}[V^2]$. At the same time, (5) can be obtained by linearizing (3) for small T_C. ∎

By combining (4) and (5), we obtain the important result:

Corollary 2: The hit probability under small-cache regime can be approximated as

$$p_{\text{hit}} \approx \frac{C}{\gamma L} \frac{\mathbb{E}[V^2]}{\mathbb{E}^2[V]} \quad (6)$$

Remark. From (6), we gain the fundamental insight that, *when the cache size is small, relatively to the catalogue size, the hit probability of LRU under SNM traffic is insensitive to the detailed shape of the popularity profile, being inversely proportional to the average content life-span L.* Moreover, the dependency of cache performance on the requests volume distribution (V) is mediated by only the first two moments of it, through the ratio $\mathbb{E}[V^2]/\mathbb{E}^2[V]$. Finally, according to (6), the hit probability increases linearly with the cache size. We will show in Sec. V that (6) also provides a satisfactory approximation for quite large cache sizes, suggesting the practical relevance of this simple formula.

3) Large-cache regime: As the cache size (C) tends to infinity, a closed-form expression for the asymptotic hit probability, denoted by $p_{\text{hit},\infty}$, can be derived from (2) by making $T_C \to \infty$.

Corollary 3: For large cache sizes,

$$p_{\text{hit},\infty} = 1 - \frac{1-\phi_V(-1)}{\mathbb{E}[V]} = 1 - \frac{1}{\mathbb{E}[V]} + \frac{\mathbb{E}[e^{-V}]}{\mathbb{E}[V]} \quad (7)$$

Observe that the dependency on the content popularity profile ($\lambda(t)$) is completely washed out in (7). Thus, the impact of temporal locality on cache performance (in particular, the popularity profile) tends to vanish as the cache size increases. This fact can be easily explained by observing that, for arbitrarily large cache sizes, the first request for a content necessarily produces a miss, whereas all subsequent requests lead to a hit.

Remark. As the cache size increases and (6) degrades in its accuracy, the hit probability tends to be affected by the detailed shape of the temporal profile as well as the distribution of the requests volume (as predicted by (2)). However, the overall impact of temporal locality on the cache performance decreases, up to the point of completely vanishing when the time scale of cache dynamics (governed by the eviction time (T_C)) becomes larger than the content life-span (L) (see (7)).

4) Extension to multi-class scenario: The single-class model in Sec. IV-A can be extended to consider the more realistic scenario in which contents are partitioned into K classes. We assume that each class is characterized by a different popularity profile ($\lambda_k(t)$), with an associated average content life-span L_k, and a request volume $V_{(k)}$, for $1 \le k \le K$. Similarly to the single-class case, we assume $\mathbb{E}[V_{(k)}] < \infty$ and we define $\phi_{V_{(k)}}(x)$ as the moment generating function for $V_{(k)}$. We can formalize the multi-class scenario by assuming that every generated content m is assigned a random mark W_m, representing the class the content belongs to, taking values in $\{1,\dots,K\}$. Assuming $\{W_m\}_m$ to be i.i.d. random variables, the total stochastic intensity of the request process at time t is given by:

$$\Lambda(t) = \sum_{m:\tau_m<t} V_m \lambda_{W_m}(t-\tau_m)$$

Under this assumption, we can state the following:

Theorem 2: Consider a cache of size C, implementing the LRU policy, and operating under a multi-class SNM model with total stochastic intensity: $\Lambda(t) = \sum_m V_m \lambda_{W_m}(t-\tau_m)$. Extending Che's approximation, the hit probability is given by:

$$p_{\text{hit}} = 1 - \sum_{k=1}^{K} \Pr\{W_1 = k\} \int_0^{\infty} \lambda_k(\tau) \frac{\phi'_{V_{(k)}}\left(-\int_0^{T_c} \lambda_k(\tau-\theta)d\theta\right)}{\mathbb{E}[V_{(k)}]} \,\mathrm{d}\tau \quad (8)$$

where T_C is the only solution to equation:

$$C = \gamma \int_0^{\infty} \left[1 - \sum_{1}^{K} \Pr\{W_1 = k\} \cdot \phi_{V_{(k)}}\left(-\int_0^{T_C} \lambda_k(\tau-\theta)d\theta\right)\right] \mathrm{d}\tau \quad (9)$$

The proof for Theorem 2 (not reported here for the sake of brevity) follows the same lines as in the proof of Theorem 1. Furthermore, when the cache size becomes small, it is possible to derive a closed-form approximation of (8) and (9):

Corollary 4: If $\mathbb{E}[V_{(k)}^2] < \infty$ for any $1 \le k \le K$, for small cache sizes the hit probability can be approximated as:

$$p_{\text{hit}} \approx T_C \sum_{k=1}^{K} \Pr\{W_1 = k\} \frac{1}{L_k} \frac{\mathbb{E}[V_{(k)}^2]}{\mathbb{E}[V_{(k)}]} \quad (10)$$

where T_C derives from equation: $C = \gamma \mathbb{E}[V] T_C$.

Remark. When cache size is small, the hit probability in the multi-class case is given by a weighted sum of contributions, related to the various classes, where each contribution is inversely proportional to the average life-span of the corresponding class and proportional to the ratio $\mathbb{E}[V_{(k)}^2]/\mathbb{E}[V_{(k)}]$.

5) Diurnal patterns and cache invariance: From our model we can derive an analytical explanation of why diurnal variations in the aggregate arrival rate of requests, such as those illustrated in Fig. 1, have no impact on the hit probability.

Diurnal variations in the intensity of the arrival process of requests can be obtained from the resulting effect of an envelope-modulation applied to all its constituent components (i.e., the shots associated to individual contents). In fact, we can obtain any desired modulation in the total intensity of the arrival process by starting from a stationary sequence of content requests, and properly diluting/densifying the associated timestamps over time, whilst preserving the ordering of the requests. To make the previous argument more rigorous, we introduce a *virtual time* function represented by a generic increasing and continuously differentiable function $w(t)$, whose first derivative $w'(t)$ is proportional to the desired instantaneous aggregate request rate at time t. Function $w(t)$ satisfies the following additional properties: $w(0) = 0$, $\lim_{t\to\infty} w(t)/t = 1$. Then we can specify a generalized SNM whereby all temporal dynamics are defined over the virtual time $w(t)$ (which replaces the original real time t). In particular, the starting time and the popularity profile of each individual content are transformed according to: $\tau_m \to w(\tau_m)$ and $\lambda(t-\tau) \to \lambda(w(t)-w(\tau))w'(t)$. In doing so, we obtain the desired effect of applying the amplitude modulation $w'(t)$ to the original process. Indeed, by construction, the average aggregate instantaneous request rate at time t becomes:

$$\lim_{\Delta t \to 0} \frac{\mathbb{E}[\sum_m \int_t^{t+\Delta t} V_m \lambda(w(t)-w(\tau_m))w'(t)dt]}{\Delta t} = \gamma \mathbb{E}[V] w'(t)$$

We can prove the following:

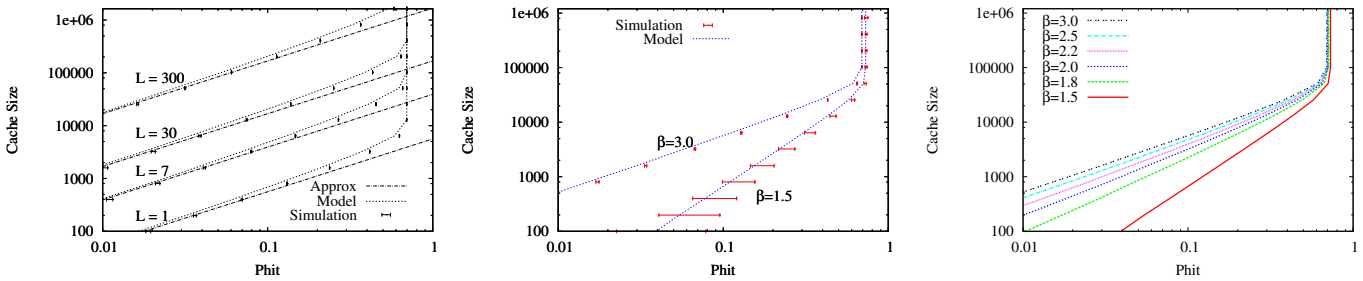
Theorem 3: Under Che's approximation, cache performance is invariant under the transformation $t \to w(t)$.

Proof: We follow exactly the same lines as in the proof of Theorem 1. In particular, the expression for p_{hit} can be obtained from (2) and (3) by substituting τ with $w(\tau)$, θ with $w(\theta)$, $d\tau \to w'(\tau)d\tau$, $d\theta \to w'(\theta)d\theta$. Then the invariance property of p_{hit} derives from the standard change of variable rule inside the integrals. ∎

Remark. Day-night fluctuations in the arrival rate of requests have no impact on cache performance, so long as such fluctuations are roughly synchronized at the ingress points of the cache network, i.e., when users reside in the same time-zone (or in few adjacent time-zones). Diurnal variations can be important in cache systems covering several time zones. In this paper we do not investigate the effects of different time-zones, because they are not observed in our data-set, and therefore cannot be validated with any reasonable level of confidence. Nonetheless, if needed, our SNM traffic (and the relative analysis) can be easily extended to incorporate "out-of-phase" fluctuations at different ingress points.

V. Numerical Evaluation

The goal of this section is two-fold. First, we assess the accuracy of the analytical model for the cache hit probability,



(a) The impact of varying the average content life-span L (expressed in days).

(b) The impact of two different values of Pareto exponent β (and the corresponding Zipf's α).

(c) The model results for various values of the Pareto exponent β.

Fig. 9: Cache performance when varying the average life-span L, in the case of a Pareto distribution (with $\beta = 3$) of requests' volume (Fig. 9a); and while varying the Pareto exponent β, in the case of average content life-span $L = 7$ days (Figs. 9b and 9c).

described in Sec. IV. Second, we exploit the insights gained from the model to better understand the performance of caching systems in the presence of temporal and geographical locality, showing that our analysis can be useful for system design and optimization. We compare the results obtained by the model against Monte-Carlo simulations of LRU, using the same synthetic SNM traffic considered in the analysis. By so doing, we are able to decouple the errors arising from modeling approximations, from those that derive from a non-perfect match between experimental (trace-driven) and synthetic traffic patterns, which have been discussed before (see Fig. 6 in Sec. III-B, and Fig. 8 in Sec. III-D). Moreover, simulating LRU under the SNM traffic model enables us to explore a much wider range of scenarios than are present in our data set, and provides us with fundamental insights into the impact of the various traffic parameters.

A. *Single-cache, single-class scenario*

We start by considering the basic case of a single cache fed by a single-class SNM traffic. We set the arrival rate of new contents (γ) to $10,000$ units per day and assume the average number of requests (V) attracted by each content to follow a Pareto distribution: $f_V(v) = \beta V_{\min}^{\beta}/v^{1+\beta}$, for $v \geq V_{\min}$[3]. The choice of a Pareto distribution for V is justified by two factors. First, previous works have shown that the aggregate requests attracted by many types of contents (including popular movies or user-generated videos) over long time periods are well described by the Zipf's law [9]. Second, a Zipf-like distribution is obtained when a large number of individual content request volumes are generated independently according to a Pareto distribution with exponent β. For the experiments presented in this section, we fix the average number of requests for each content to $\mathbb{E}[V] = 3$. Since the shape of the popularity profile has been shown to have a negligible impact on the resulting cache performance (see Sec. III and IV), unless otherwise specified we assume a uniform popularity profile, with average life-span L. Finally, for the results obtained by simulation, we show the error bars corresponding to 95% confidence intervals.

Fig. 9a shows the required cache size to achieve a given hit probability, for different values of L. We observe an almost perfect match between simulation results (the horizontal error-bars appear as points) and the model prediction from (2) (dotted lines). We find that our small-cache approximation (4) (solid line) is very accurate for a wide range of values of p_{hit}. As expected, cache performance is deeply impacted by the average life-span of contents (L): as suggested by the closed-form approximation (4), for a given cache size, the hit probability is roughly inversely proportional to the average life-span (L).

To investigate the impact of the distribution of the number of requests attracted by contents (V), Figs. 9b and 9c show the results obtained when varying the value of the Pareto exponent β. Comparing the analytical prediction (2) against simulations for two extreme values of β, in Fig. 9b, we observe that the model is very accurate. Fig. 9c reports the results for a wider range of β; for the sake of clarity, here we omit the simulation results, since we observed a strong agreement between model and simulation results in all cases.

As expected in general the distribution of the number of requests attracted by contents (V), may play an important rule on the cache performance (i.e., the cache size required to achieve a given hit probability); the performance of the caching system benefits from making the popularity distribution less and less skewed (i.e., by decreasing β). Note, however that the *impact on cache performance of the specific β is fairly limited as long as $\beta > 2$ (i.e., the variance of the number of content requests keeps finite).* This is in sharp contrast to results obtained under IRM traffic, where a small change of the Zipf's exponent has a huge impact on cache performance [9]. As a consequence, *under SNM, a precise characterization of popularity distribution parameters is not that important to predict cache performance (as long as the number of requests attracted by contents has a finite variance).*

B. *Single-cache, multi-class scenario*

We now move on to the case of a single cache fed by a multi-class SNM traffic, with the goal of understanding the impact on cache performance of a mixture of highly heterogeneous contents characterized by different degrees of temporal locality. This is indeed the kind of traffic that we observe in a real network, as we found in our data set (see Table III). In particular, we consider the 6 classes of contents listed in Table VI, whose parameters have been chosen to reasonably match a realistic scenario (see Sec. II). Class 0 collects unpopular contents with request volumes smaller than 10. Classes 1–5 correspond to popular contents

[3] Note that the second moment of V is finite for $\beta > 2$

Class	L (days)	$E[V]$	$V_{\max}$	β	Scen. 1	Scen. 2	Scen. 3
0	500	1.61	10	2.5	0.85	0.85	0.85
1	2	83.33	∞	2.5	0.00	0.00	0.01
2	7	75.00	∞	2.5	0.00	0.02	0.02
3	30	66.66	∞	2.5	0.02	0.02	0.02
4	100	50.00	∞	2.5	0.02	0.02	0.02
5	1000	50.00	∞	2.5	0.11	0.09	0.08

TABLE VI: Content class parameters and their composition for each multi-class scenario.

having different degree of temporal locality, with average life-span (L) ranging from a few days (Class 1) to several years (Class 5). The different values for the average number of requests attracted by contents in these classes reflect the observations from our traces (see Table III).

In order to understand the impact of different traffic mixes, we consider 3 traffic scenarios in which we vary the proportion of each class of contents (i.e., the probability $\Pr\{W_1 = k\}$ that a new content belongs to a given class), as reported in the last 3 columns of Table VI. Note that Class 1 is missing in both *Scenario 1* and *Scenario 2*, whereas Class 2 is missing only in *Scenario 1*. For all scenarios the arrival rate of new contents is set to $\gamma = 10^5$ contents/day. Finally, an exponential popularity profile is chosen for all the classes.

Fig. 10 reports the cache performance under the three considered scenarios, as obtained by (8). We observe that *the presence of just a small fraction of highly cacheable contents* (in *Scenario 3* only 3% of the contents belong to either Class *1* or *2*) *has a huge beneficial impact on the hit probability, especially with small caches.* Even for medium-size caches the gain is very significant: for example, in the case of $C = 10,000$, the hit probability goes from 5% (*Scenario 1*) to about 20% (*Scenario 3*).

Previous results suggest that contents characterized by high temporal locality, although few in number, do play the major role in the resulting hit probability. This fact also suggests that, *when the cache size is limited, it may be convenient to devote the entire cache space only to highly cacheable contents (i.e., contents with large volumes and significant temporal locality), and to forbid other contents from entering the cache.* This strategy minimizes the probability of evicting from the cache contents with a high temporal locality in their request pattern, to let room to an unpopular content which will likely not be requested again while being cached (hence storing this content in the cache is useless). To check the extent to which this assertion is valid, we modified the classical LRU caching strategy (and the corresponding analysis) such that contents belonging to specified classes are never cached (notice that, by so doing, all requests for filtered contents deterministically produce a miss). The extension of the analysis to compute the resulting hit probability on this LRU variant is rather straightforward, hence we omit the details here. Under *Scenario 3*, Fig 11 compares the performance of LRU against the performance of LRU-0 and LRU-(0+5), which do not cache contents of class *0* and of both classes *0* and *5*, respectively. Observe that, *when the cache size is limited, a significant performance improvement is achieved by filtering out contents that are either unpopular (class 0) or popular but long-lived (class 5).* For example, the adoption of LRU-(0+5) leads to a reduction of more than one order of magnitude in the cache size needed to achieve $p_{\text{hit}} = 0.1$, with respect to LRU. Finally, as expected, filtering out contents when the cache size increases must at some point become deleterious, since filtered contents lead to a miss in the cache. This is confirmed by the intersection between the curves in Fig 11.

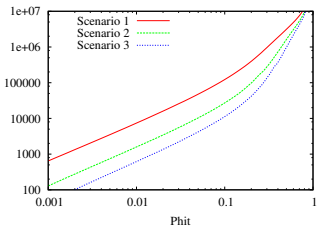


Fig. 10: Cache performance for different traffic scenarios.

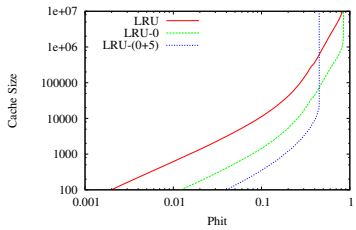


Fig. 11: Cache performance when; i) no filtering, ii) filtering Class *0*, iii) filtering Classes *0* and *5*.

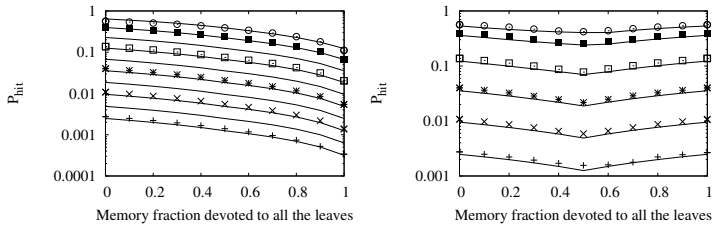
The practical implementation of filters to detect unpopular/long lived contents raises issues that go beyond the scope of this paper. Here we limit ourselves to mentioning that content classification can be accomplished either by exploiting a-priori information about the content, such as the category, the producer etc., or by employing blind online techniques to infer the instantaneous request rate subject to the history of requests [26].

C. Multi-cache, single-class scenario

We now consider a simple cache network with a tree structure. In addition to assessing the accuracy of the improved approximation described in Appendix B, this scenario permits us to understand the impact of geographical locality on cache performance. In more detail, we consider a two-layer cache network composed by 8 leaves and one root (plus an additional repository above the root). As in the standard Edge CDN architectures [2], [18], content requests arrive at the leaves, and misses are forwarded to the root. Here, we focus on two extreme traffic scenarios: i) an *unlocalized* scenario, in which content requests are equally likely to arrive at any of the leaves (independently at random), and ii) a *fully localized* scenario, in which each leaf receives the requests for just a subset of the entire catalog (i.e., each newly introduced content is statically assigned to a distinct leaf, which will receive all its associated requests). For both scenarios, we consider a single-class SNM with the following parameters: requests volumes V are Pareto-distributed with $\mathbb{E}[V] = 3.0$ and $\beta = 2.5$; the popularity profile is exponential with average life-span $L = 7$ days; the aggregate arrival rate of new contents (γ) is set to $10,000$ contents per day.

Being conscious that the extreme scenarios proposed above are oversimplified and may look somehow artificial, we emphasize that our goal here is to understand the potential impact of geographical locality on the overall performance of a caching system. Hence, by considering the two extreme cases above, we can evaluate the whole range of possible behavior of the system under intermediate (more realistic) traffic patterns.

Fig. 12 reports the global hit probability (i.e. either at any leaf or at the root cache), for the *unlocalized* (Fig. 12a) and *fully localized* (Fig. 12b) scenario, as function of the fraction of total storage capacity assigned to the leaves (i.e., we assume that the total capacity of all caches is kept constant). We show both the results obtained analytically with the improved approximation explained in Appendix B, and simulation results obtained under the same SNM. In both scenarios, the analytical predictions (lines) match very well simulation results (marks). Beyond proving the accuracy of



(a) Unlocalized scenario. (b) Localized scenario.

Fig. 12: Hit probability for different cache size under different traffic localization scenarios. Lines refer to the analytical model, points to the simulation results. Caches sizes are (from bottom to top) 100, 400, 1600, 6400, 25600, 51200 contents.

the model, some interesting insights at system level can be obtained from the plots in Fig. 12. *When no geographical locality is present* (Fig. 12a), *the maximum hit probability is achieved when the whole storage capacity is located in a single cache (the root).* This can however result in longer access delays for the users. *Instead, when traffic is strongly localized,* (Fig. 12b), *by increasing the cache size of the leaves (up to the point at which all storage capacity is assigned to the leaves) we jointly maximize the cache hit probability while reducing the access delay.* Interestingly, in this case the same maximum hit probability is also achieved by putting all storage in the root, although this would be detrimental in terms of delay.

At last we emphasize that geographical locality plays a similar role also under a multi-class scenario, for which do not report results due to space constraints.

VI. Conclusions

The Shot Noise Model provides a simple, flexible and accurate approach to describing the temporal and geographical locality found in Video-on-Demand traffic, allowing us also to develop accurate analytical models of cache performance. From the point of view of system design, our main findings are: i) cache performance can significantly benefit from the presence of even a relatively small portion of highly cacheable (popular and local) contents (especially when caches are small); ii) geographical locality plays also an important role in the dimensioning of distributed caching systems and should not be neglected; iii) the overall impact on cache performance of the distribution of the number of requests attracted by the contents (and the corresponding rank distribution) is significantly mitigated by temporal locality with respect to traditional stationary models (e.g., IRM); iv) especially when caches are small, performance can be significantly improved by restricting access into the cache only to contents which are highly cacheable. This can be obtained either exploiting a priori information about the contents' nature and popularity profiles, or by measuring the content instantaneous popularity.

References

[1] "Cisco Visual Networking index: forecast and methodology, 2012–2017," White paper, 2013.

[2] R. K. S. Erik Nygren and J. Sun, "The Akamai network: A platform for high-performance Internet applications," *SIGOPS OSR*, 2010.

[3] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, and N. H. Briggs, "Networking named content," in *ACM CoNEXT*, 2009.

[4] W. Jiang, S. Ioannidis, L. Massoulié, and F. Picconi, "Orchestrating massively distributed CDNs," in *ACM CoNEXT*, 2012.

[5] I. Poese, B. Frank, G. Smaragdakis, S. Uhlig, A. Feldmann, and B. Maggs, "Enabling content-aware traffic engineering," *ACM SIGCOMM CCR*, Sep. 2012.

[6] M. Xie, I. Widjaja, and H. Wang, "Enhancing cache robustness for content-centric networking," in *IEEE INFOCOM*, 2012.

[7] M. Gallo, B. Kauffmann, L. Muscariello, A. Simonian, and C. Tanguy, "Performance evaluation of the random replacement policy for networks of caches," in *ACM SIGMETRICS*, 2012.

[8] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, "Impact of traffic mix on caching performance in a content-centric network," in *NOMEN Workshop*, 2012.

[9] C. Fricker, P. Robert, and J. Roberts, "A versatile and accurate approximation for LRU cache performance," in *ITC*, 2012.

[10] E. Coffman and P. Denning., *Operating Systems Theory.* Englewood Cliffs (NJ): Prentice-Hall, 1973.

[11] M. C. R. Fonseca, V. Almeida and B. Abrahao, "On the intrinsic locality of web reference streams," in *IEEE INFOCOM*, 2003.

[12] V. Almeida, A. Bestavros, M. Crovella, and A. de Oliveira, "Characterizing reference locality in the www," in *IEEE PDIS*, 1996.

[13] S. Jin and A. Bestavros, "Sources and characteristics of web temporal locality," in *IEEE MASCOTS*, 2000.

[14] R. Crane and D. Sornette, "Robust dynamic classes revealed by measuring the response function of a social system," *PNAS*, vol. 105, no. 15469, 2008.

[15] K. Y. Kamath, J. Caverlee, K. Lee, and Z. Cheng, "Spatio-temporal dynamics of online memes: a study of geo-tagged tweets," in *ACM WWW*, 2013.

[16] M. M. S. Scellato, C. Mascolo and J. Crowcroft, "Track globally, deliver locally: improving content delivery networks by tracking geographic social cascades," in *ACM WWW*, 2011.

[17] A. Brodersen, S. Scellato, and M. Wattenhofer, "Youtube around the world: geographic popularity of videos," in *ACM WWW*, 2012.

[18] Q. Huang, K. Birman, R. van Renesse, W. Lloyd, S. Kumar, and H. C. Li, "An analysis of facebook photo caching," in *ACM SOSP*, 2013.

[19] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *IEEE INFOCOM*, 2010.

[20] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. K. Ramakrishnan, "Optimal content placement for a large-scale vod system," in *ACM CoNEXT*, 2010.

[21] H. Abrahamsson and M. Nordmark, "Program popularity and viewer behaviour in a large tv-on-demand system," in *ACM IMC*, 2012.

[22] A. Mahanti, D. Eager, and C. Williamson, "Temporal locality and its impact on web proxy cache performance," *Perf. Eval.*, vol. 42(23), 2000.

[23] J. Møller, "Shot noise Cox processes," *Advances in Applied Probability*, vol. 35, no. 3, 2003.

[24] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi, and S. Niccolini, "Temporal locality in today's content caching: Why it matters and how to model it," *SIGCOMM CCR*, 2013.

[25] H. Che, Y. Tung, and Z. Wang, "Hierarchical web caching systems: modeling, design and experimental results," *IEEE JSAC*, vol. 20, no. 7, Sep. 2002.

[26] P. R. Jelenković and A. Radovanović, "The persistent-access-caching algorithm," *Random Struct. Algorithms*, vol. 33, no. 2, Sep. 2008.

[27] R. W. Wolf, "Stochastic modeling and the theory of queues," *Prentice Hall*, 1989.

[28] E. Rosensweig, J. Kurose, and D. Towsley, "Approximate models for general cache networks," in *IEEE INFOCOM*, 2010.

[29] N. Fofack, P. Nain, G. Neglia, and D. Towsley, "Analysis of TTL-based cache networks," in *VALUETOOLS*, 2012.

Appendix A
Proof of Theorem 1

Proof: By adopting Che's approximation, we assume that the cache eviction T_C time is constant and independent of the considered content. We express the probability of finding a given content m in the cache at time t, conditionally on its starting time (τ_m) and average request volume (V_m), as:

$$p_{\text{in}}(t \mid \tau_m, V_m) = 1 - e^{-V_m \int_{t-T_C}^{t} \lambda(\theta-\tau_m)d\theta} \tag{11}$$

since, under LRU, the considered content is found in the cache at time t, iff it attracted at least one request in the time interval $[t - T_C, t]$. Unconditioning (11) with respect to V_m, and recalling that V_m are i.i.d. as V, we obtain:

$$p_{\text{in}}(t \mid \tau_m) = \mathbb{E}_V\left[1 - e^{-V \int_{t-T_C}^{t} \lambda(\theta-\tau_m)d\theta}\right] = 1 - \phi_V\left(-\int_{t-T_C}^{t} \lambda(\theta-\tau_m)d\theta\right)$$

To evaluate the probability of finding content m in the cache at time t, we uncondition the above expression with respect to τ_m. To do this, recall that in a Poisson process, conditionally over the number of points falling in the interval $[0,t)$, each point is uniformly distributed over the considered interval, independently of other points. Hence, the distribution of τ_m is uniform in the interval $[0,t)$, and we obtain:

$$p_{\text{in}}(t) = \frac{1}{t}\int_0^t 1-\phi_V\left(-\int_{t-T_C}^t \lambda(\theta-\tau)d\theta\right)d\tau \qquad (12)$$

Now, as in the standard IRM, for a sufficiently large t, we can assume that the cache is completely filled with contents introduced before t, and the number of contents in the cache is exactly equal to its size. Therefore we can write:

$$C = \sum_m [\mathbb{I}_{\{\text{content } m \text{ in cache at time } t | \tau_m \leq t\}} \mathbb{I}_{\tau_m \leq t}]$$

where the sum extends over all contents in the infinite content catalogue. Averaging both terms we obtain:

$$C = \sum_m \mathbb{E}[\mathbb{I}_{\{m \text{ in cache at } t | \tau_m \leq t\}} \mathbb{I}_{\tau_m \leq t}] = p_{\text{in}}(t) \sum_m \mathbb{E}[\mathbb{I}_{\tau_m \leq t}] \qquad (13)$$

Recalling that the average rate at which new contents are introduced is γ, by combining (12) with (13), we can express the size of the cache C as:

$$C = \left(\sum_m \frac{\mathbb{E}[\mathbb{I}_{\tau_m<t}]}{t}\right)\int_0^t 1-\phi_V\left(-\int_{t-T_C}^t \lambda(\theta-\tau)d\theta\right)d\tau = \gamma\int_0^t 1-\phi_V\left(-\int_{t-T_C}^t \lambda(\theta-\tau)d\theta\right)d\tau \qquad (14)$$

Eq. (14), proving (3), must be solved (numerically) to evaluate the eviction time (T_C) for a given cache size C. Furthermore, (14) provides an interesting insight into the cache behavior: *for a given eviction time T_C, the cache size C is proportional to the rate at which new contents are introduced (γ).*

Now we turn our attention to the hit probability. Assume that a request R_t arrives at the cache at time t for content m of parameters (τ_m, V_m). By definition, R_t generates a cache hit iff the content is found in the cache. Therefore, as consequence of the Lack of Anticipation (LAA) property [27] of the request process for content m, the hit probability experienced by request R_t is:

$$p_{\text{hit}}(t \mid \tau_m, V_m) = p_{in}(t \mid \tau_m, V_m)$$

Now, when unconditioning $p_{\text{hit}}(t \mid \tau_m, V_m)$ with respect to (τ_m, V_m), we have to carefully account for the fact that contents are not uniformly requested. Observe that the instantaneous request at which cache requests rate for a specific content m is given $V_m\lambda(t-\tau_m)$. Thus, we can interpret:

$$V_m\lambda(t-\tau_m)\cdot p_{\text{hit}}(t \mid \tau_m, V_m)$$

as the hit rate generated by content m. Summing up all contents, we can express $p_{\text{hit}}(t)$ as the ratio between average global cache hit-rate and average global request rate. It turns out that:

$$p_{\text{hit}}(t) = \frac{\mathbb{E}[\sum_m V_m\lambda(t-\tau_m)\cdot p_{\text{hit}}(t \mid \tau_m, V_m)]}{\mathbb{E}[\sum_m V_m\lambda(t-\tau_m)]}$$

Recalling (11) we have for $t \geq T_C$:

$$p_{\text{hit}}(t) = \frac{\gamma\int_0^t \mathbb{E}_V\left[V\lambda(t-\tau)\left(1-e^{-V\int_{t-T_C}^t \lambda(\theta-\tau)d\theta}\right)\right]d\tau}{\gamma\mathbb{E}[V]\int_0^t \lambda(t-\sigma)d\sigma} = \int_0^t \lambda(t-\tau)\left(1-\frac{\phi_V'\left(-\int_{t-T_C}^t \lambda(\theta-\tau)d\theta\right)}{\mathbb{E}[V]\int_0^t \lambda(t-\sigma)d\sigma}\right)d\tau$$

Substituting $\alpha = t-\tau$, $\beta = t-\theta$, $\zeta = t-\sigma$ we get:

$$p_{\text{hit}}(t) = \int_0^t \lambda(\alpha)\left(1-\frac{\phi_V'\left(-\int_0^{T_C} \lambda(\alpha-\beta)d\beta\right)}{\mathbb{E}[V]\int_0^t \lambda(\zeta)d\zeta}\right)d\alpha \qquad (15)$$

Thanks to the integrability property of $\lambda(t)$, (2) is obtained by letting $t \to \infty$ in (15). ■

APPENDIX B
LRU IN CACHE NETWORKS

We now show how our analysis of LRU policy under SNM can be extended to the case of a network of caches, whereby misses are forwarded to other caches, according to pre-established routes, possibly ending up at a repository storing the entire catalogue. In doing so, we will propose an improved approximation with respect to the one proposed in [28], which is based on the simplifying assumption that the arrival process of requests for a given content at any cache is Poisson. For simplicity, we will consider only networks implementing the so-called leave-copy-everywhere replication strategy, according to which a copy of a requested content is inserted in all caches traversed by the request. Moreover, we will restrict ourselves to the case of tree-like topologies [4]. Requests arrive initially at the leaves of the tree. Whenever a content is not found at a leave, it is forwarded to the parent node. We assume that there exists a repository storing the entire catalogue above the root of the tree.

A. The Poisson approximation

For any cache c in the network, we denote by $\mathcal{C}(c)$ the set of children of c (caches) in the tree. Let $\mathcal{F}$ be the set of caches corresponding to the leaves of the tree. We denote by $\mathcal{F}(c)$ the subset of $\mathcal{F}$ corresponding to the descendants of c in the tree. According to the model proposed in Sec. III-C, at time t, the arrival rate of requests for content m at leaf node $f \in \mathcal{F}$ is given by:

$$\lambda^{(f)}(t \mid V_m, \tau_m) = V_{m,f}\lambda_m(t-\tau_m) = V_m p_{m,f}\lambda_m(t-\tau_m)$$

where V_m is the total request volume produced by content m and τ_m is the time at which content m is introduced into the catalogue. Recall also that $p_{m,f}$ represents the probability that a request for content m enters the network at cache f. By construction, $\sum_{f\in\mathcal{F}} p_{m,f} = 1$. We observe that each leaf can be independently analyzed using (2) and (3).

Consider now a non-leaf node c; the intensity of the arrival process of requests for content m at c at time t is given by:

$$\lambda_m^{(c)}(t \mid V_m, \tau_m) = \sum_{c'\in\mathcal{C}(c)} \lambda_m^{(c')}(t \mid V_m, \tau_m)(1-\mathbb{I}_{\{m\in c'\}}(t \mid V_m, \tau_m))$$

[4]The extension of our analysis to general mesh networks can be carried out in a similar way as proposed in [28] under the Poisson approximation. This extension is conceptually very simple, but requires a global multi-variable fixed point procedure to solve the entire system.

where $\mathbb{I}_{\{m\in c'\}}(t \mid V_m, \tau_m)$ is the indicating function corresponding to the event $\{m \in c' \text{ at time } t \mid V_m, \tau_m\}$. As such, $\lambda_m^{(c)}(t \mid V_m, \tau_m)$ turns out to be a stochastic variable, because its value dependent on the state of caches in $\mathcal{C}(c)$. This implies that the arrival process of requests at cache c is not an inhomogeneous Poisson process. The expectation of $\lambda_m^c(t \mid V_m, \tau_m)$ can be computed as:

$$\overline{\lambda}_m^{(c)}(t \mid V_m, \tau_m) = \mathbb{E}[\lambda_m^{(c)}(t \mid V_m, \tau_m)] = \\ = \sum_{c'\in\mathcal{C}(c)} \mathbb{E}[\lambda_m^{(c')}(t \mid V_m, \tau_m)](1 - p_{\text{in}}^{(c')}(t \mid V_m, \tau_m)) \quad (16)$$

where $p_{\text{in}}^{(c')}(t \mid V_m, \tau_m)$ is the probability that content m (with attributes (τ_m, V_m)) is cached in c' at time t.

The standard approximation to (16) would be to replace $\lambda_m^c(t \mid V_m, \tau_m)$ with its expectation $\overline{\lambda}_m^{(c)}(t \mid V_m, \tau_m)$ at all nodes which are not leaves, i.e., to approximate the arriving process of requests for content m at c, with an inhomogeneous Poisson process whose instantaneous intensity at time t is equal to the average intensity of the actual process at time t, and then independently solve each cache in isolation using the single-cache analysis.

At last, approaches that go beyond the Poisson approximation have been recently proposed in [29]. These approaches attempt to better characterize the cache miss stream. However, they rely heavily on the assumption that the request arrival process for a given content at a cache is a stationary (renewal) process, and are therefore not applicable to our case.

B. Improved approximation

Our basic idea is to approximate the correlation between the states of neighboring caches, which is totally neglected under the Poisson approximation. This can be done by distinguishing between the hit probability ($p_{\text{hit}}^{(c)}(t \mid V_m, \tau_m)$) and the probability of finding a given content m in the cache at time t ($p_{\text{in}}^{(c)}(t \mid V_m, \tau_m)$). Note that, while the hit probability is implicitly conditioned to the event that a request for m arrives at cache c at time t, the probability of the content being present in the cache is not. By virtue of the lack of anticipation property [27], the above two probabilities would be equal if the arrival process was an inhomogeneous Poisson process. However, the arrival process at non-leaf nodes is not Poisson, hence the above two probabilities can be different at non-leaf nodes.

To approximately evaluate $p_{\text{hit}}^{(c)}(t \mid V_m, \tau_m)$, we focus on a request for content m arriving at cache c at time t from a given child node c'. We observe that such a request can arrive at time t at cache c, only if content m is not stored in cache c' at t^-. This implies that no request for m can have arrived to c' in the interval $[t - T_C^{(c')}, t]$ (otherwise content m would be stored in cache c' at time t^-). Therefore, a fortiori, no request for m, coming from c', can have arrived at cache c in the same interval. Indeed, content m is found in cache c at time t by a request arriving from c' if and only if either (i) at least one request arrived at cache c within the interval $[t-T_C^{(c)}, t-T_C^{(c')}]$ from any child node, including c' (this case is considered provided that $T_C^{(c)} > T_C^{(c')}$); or (ii) at least one request arrived at cache c within the interval $[t-T_C^{(c')}, t]$ from $c'' \neq c'$, i.e., from caches different from c' (since we know that no request can arrive at c from c' during this interval).

During both intervals considered above, the arrival process of requests at cache c from any child node c' is not Poisson (but depends on the unknown state of the child node), and lacking a better approach, we resort to approximating it by a Poisson process with the expected intensity. By so doing we can compute the conditioned hit probability $p_{\text{hit}}(t \mid V_m, \tau_m, c')$ for requests coming from c' as:

$$p_{\text{hit}}(t \mid V_m, \tau_m, c') \approx 1 - e^{-\int_{t-T_C^{(c)}}^{t-\min(T_C^{(c')}, T_C^{(c)})} \overline{\lambda}_m^{(c)}(\tau \mid V_m, \tau_m)\, \mathrm{d}\tau} \cdot \\ \prod_{c''\in\mathcal{C}(c)\setminus c'} e^{-\int_{t-\min(T_C^{(c')}, T_C^{(c)})}^{t} \overline{\lambda}_m^{(c'')}(\tau \mid V_m, \tau_m)\left(1 - p_{\text{in}}^{(c'')}(\tau \mid V_m, \tau_m)\right) \mathrm{d}\tau}$$

Unconditioning with respect to c' (i.e., by properly taking into account the fraction of requests for m arriving at c at time t from each child), we obtain an approximate expression for the overall hit probability of content m at cache c at time t (we omit the details of this unconditioning). The above reasoning cannot be applied to the computation of $p_{\text{in}}^{(c)}(t \mid V_m, \tau_m)$, for which we resort to the standard Poisson approximation.