Performance Evaluation 113 (2017) 68–82

Contents lists available at ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

Optimization of caching devices with geometric constraints

Konstantin Avrachenkov [a], Xinwei Bai [b], Jasper Goseling [b,*]

[a] INRIA Sophia Antipolis, France
[b] Stochastic Operations Research, University of Twente, The Netherlands

ARTICLE INFO

Article history:
Received 11 February 2016
Received in revised form 3 January 2017
Accepted 4 May 2017
Available online 15 May 2017

Keywords:
Caching
Wireless networks
Stochastic geometry

ABSTRACT

It has been recently advocated that in large communication systems it is beneficial both for the users and for the network as a whole to store content closer to users. One particular implementation of such an approach is to co-locate caches with wireless base stations. In this paper we study geographically distributed caching of a fixed collection of files. We model cache placement with the help of stochastic geometry and optimize the allocation of storage capacity among files in order to minimize the cache miss probability. We consider both per cache capacity constraints as well as an average capacity constraint over all caches. The case of per cache capacity constraints can be efficiently solved using dynamic programming, whereas the case of the average constraint leads to a convex optimization problem. We demonstrate that the average constraint leads to significantly smaller cache miss probability. Finally, we suggest a simple LRU-based policy for geographically distributed caching and show that its performance is close to the optimal.

1. Introduction

We consider caching of a collection of files by a set of geographically distributed storage devices with wireless communications capabilities and random network coding. Clients can retrieve cached data from all devices that are within its connectivity radius. Since the caching devices have limited storage capacity, not all files can be stored in all caches. Therefore, there is a positive probability that a file that is requested by a client cannot be retrieved from the caching devices that are within range and a cache miss occurs. The general aim of this paper is to optimize the cache allocation so to minimize the cache miss probability.

It has been recently advocated that in large communication systems it is beneficial both for the users and for the network as a whole to store content closer to users. This idea can be realized by Information Centric Networking (ICN), a new paradigm for the network architecture where the data is addressed by its name or content directly rather than by its physical location. There is no predefined location for the data in ICN and the content is naturally cached along the retrieval path. Examples of the ICN architecture are CCN/NDN [1], DONA [2] and TRIAD [3]. Our results can be useful for the design of the wireless networks with the ICN architecture in which case cellular base stations also serve as caches. Wireless sensor networks represent another potential application of our results. Sensors have severe limitation on both memory and transmission capability. It might be useful for sensors to have access to some aggregated characteristics in addition to the local ones. In such a case, our results provide optimal distributed allocation of the aggregated characteristics.

Let us elaborate on the problem formulation in further details. Storage (or caching) devices are placed in the plane according to a homogeneous spatial Poisson process. The homogeneous spatial Poisson process is accepted for modeling the location of base stations providing a good compromise between realistic representation of the wireless network and

* Corresponding author.
E-mail addresses: k.avrachenkov@sophia.inria.fr (K. Avrachenkov), x.bai@utwente.nl (X. Bai), j.goseling@utwente.nl (J. Goseling).

http://dx.doi.org/10.1016/j.peva.2017.05.001
0166-5316/

mathematical tractability [4–6]. For some cases, e.g., for Sydney base station network [7], it has been shown that the spatial homogeneous Poisson process represents very well the distribution of base stations. In other cases, a non-homogeneous Poisson process can be more appropriate for modeling the distribution of base stations. In fact, some results of the present work can be extended to the case of non-homogeneous Poisson process and we discuss such extensions later in the paper. The size of the file catalog is finite and fixed. A client will request one of the files from the catalog at random according to a known file popularity distribution that is the same for all clients. In particular, for numerical illustration purpose we will consider the case that file popularities follow a Zipf distribution. For the sake of tractable performance evaluation analysis, we make a technical assumption that files consist of the same number of chunks of a fixed size. We suggest to use random linear network coding, in which case linear combination of chunks can be stored in the caching devices. As was shown in [8], the network coding based allocation strategy outperforms a strategy without coding for a wide range of performance measures and any spatial distribution of caches.

Our interest in the current paper is in the case when the caches are reachable only within a fixed distance to the client. This is a standard model in wireless networks which gives high level but still quite accurate representation of a wireless connection [4,5]. Our goal is to minimize the cache miss probability, which is the probability that a client cannot get the requested file from the caches within range. Since the probability of not recovering a file from coded chunks is negligible in comparison to the overall cache miss probability, we concentrate solely on the calculation of the cache miss probability and on the optimization of the system with respect to this metric.

We have multiple files and a limited memory in each storage device. Thus, the question is how many linear combinations of each file to store in a particular storage device. Initially, we consider the case when we make the same allocation in all caches, *i.e.*, each cache stores the same number of linear combinations of each file. As a consequence we guarantee a capacity constraint on each individual cache. We formulate an optimization problem with a non-convex objective function and linear constraints. We demonstrate that this problem is a generalization of an unbounded knapsack problem [9]. In particular it is a separable nonlinear integer program, which can be solved using dynamic programming. In addition to providing a formal statement of this result, we give exact closed form results for some special cases of the problem as well as insight into the structure of the solution in the general case.

The above formulation leads to the same allocation in each storage device, which likely leads to inefficient memory utilization and to the lack of file diversity. Thus, we then turn our attention to a relaxation of the problem in which, instead of imposing a hard capacity constraint on each of the caches, we require that the average storage space used in the caches is upper bounded. In particular, we consider cache allocation strategies in which the number of linear combination to store for a file in a cache is a random variable. The number of such combinations is independently and identically decided for each cache. We impose an average capacity constraint on the number of chunks stored in a caching device, where the average is over the caching devices. We analyze the resulting optimal strategy for the case when files consist of a single piece and show that the performance under an average capacity significantly outperforms the optimal performance under a per cache capacity constraint.

Finally, we consider a dynamic scenario when the clients arrive over time. We study two LRU-based caching policies, cooperative and fully distributed. Both policies demonstrate that performance is not far from the optimal one and that there is a small loss of efficiency in the fully distributed case compared to the cooperative case. This indicates that a simple distributed LRU-based caching policy can be safely deployed in practice for geographically distributed caches. Also, it indicates that our results on the optimal placement policies can provide insight into the performance in the dynamic setting.

Let us outline the organization of the paper. In Section 3 we define the model, discuss the constraints and optimization criterion. The problem with per cache constraints is analyzed in Section 4. In particular, we provide structural insight into the optimal storage allocation strategy and show that the problem is a generalization of the unbounded knapsack problem and can be solved by dynamic programming approach. Then, in Section 5 we introduce the average constraint, which makes memory usage more efficient and increases file diversity. In an important particular case we are able to solve the average constraint problem in a closed form. In Section 6 we present distributed and cooperative LRU-based policies. In Section 7 we demonstrate that the performance of the distributed LRU-based policy is not far from the optimal performance. The numerical results of Section 7 also confirm that the average constraint in comparison with the per cache constraint, brings improved efficiency and file diversity. Finally, in Section 8 we provide a discussion of our results and an outlook on future research.

2. Related work

Literature on caching is vast. Therefore, we limit our discussion to work on caching which we feel is most relevant to the present work. The application of network coding for distributed storage is studied in [10] and in [11,12] specifically for the case of content distribution in wireless networks. The use of coding was also explored in [13] where it was shown how to efficiently allocate the data at caches with the aim of ensuring that any sufficiently large subset of caches can provide the complete data. The difference with the current work is that we are taking the geometry of the deployment of the storage devices into account. In [14], see also [15], coding strategies for networks of caches are presented, where each user has access to a single cache and a direct link to the source. It is demonstrated how coding helps to reduce the load on the link between the caches and the source. Note that we assume that different transmissions from caches to the clients are orthogonal, for instance by separating them in time or frequency. In [16] the impact of non-orthogonal transmissions is considered and