Leveraging Big Data Analytics for Cache-Enabled Wireless Networks

Manhal Abdel Kader[◇], Ejder Baştuğ[◇], Mehdi Bennis[⋆], Engin Zeydan[○],
Alper Karatepe[○], Ahmet Salih Er[○] and Mérouane Debbah[◇,†]

[◇]Large Networks and System Group (LANEAS), CentraleSupélec, Université Paris-Saclay, Gif-sur-Yvette, France
[⋆]Centre for Wireless Communications, University of Oulu, Finland
[○]AveaLabs, Istanbul, Turkey
[†]Mathematical and Algorithmic Sciences Lab, Huawei France R&D, Paris, France
{ejder.bastug, merouane.debbah}@centralesupelec.fr, bennis@ee.oulu.fi,
{engin.zeydan, alper.karatepe, ahmetsalih.er}@avea.com.tr, manhalak@gmail.com

***Abstract*—While $5G$ wireless networks are expected to handle the ever growing data avalanche, classical deployment/optimization approaches such as hyper-dense deployment of base stations or having more bandwidth are cost-inefficient, and are therefore seen as stopgaps. In this regard, context-aware approaches which exploits human predictability, recent advances in storage, edge/cloud computing and big data analytics are needed. In this article, we approach this problem from a proactive caching perspective where gains of cache-enabled base stations in $5G$ wireless are studied. In particular, huge amount of real data from a telecom operator in Turkey is collected/processed on a big data platform, and an analysis is carried out for content popularity estimation for caching, aiming to improve users' experience in terms of request satisfactions and offloading the backhaul. Subsequently, with this mobile traffic data collected from many base stations within several hours of time interval and the estimation of content popularity via machine learning tools, we investigate the gains of proactive caching via numerical simulations. The results show that proactive caching fulfils 100% of user request satisfaction and offloads 98% of the backhaul, in a setting of 16 base stations with 15.4 Gbyte of storage size (87% of the total catalog size) and 10% of content ratings.**

***Index Terms*—proactive caching, content popularity estimation, big data, machine learning, 5G cellular networks**

I. Introduction

Mobile cellular networks are nowadays facing an exponential wireless data traffic, forcing mobile operators to operate their ever-growing networks in a more complex manner. The next generation 5G wireless networks aim to fulfil this demand i.e., by device-to-device communications, edge caching (namely caching at base stations and user terminals), massive multiple-input multiple output massive multiple-input multiple-output (massive-MIMO), ultra-dense networks and millimetre wave communications (see [1] and references therein). In fact, continuous efforts for improvement of spectral efficiency and maturity of air interface in current standards (i.e., LTE-Advanced) say that no major gains can be expected in spectral efficiency, thus novel approaches are urgently needed.

This research has been supported by the ERC Starting Grant 305123 MORE (Advanced Mathematical Tools for Complex Network Engineering), the SHARING project under the Finland grant 128010, TUBITAK TEYDEB 1501 project grant (numbered 9120067) and the project BESTCOM.

Indeed, today's smart phones with anywhere at any-time connectivity, mobile video streaming, resource-intensive and over-the-top (OTT) applications, communications in diverse domains (e.g., machine-to-machine communications, smart home, healthcare, connected cars, etc.) with many characteristics (i.e., structured/non-structured) participate to this traffic explosion and falls into the framework of *Big Data* (see [2] for a recent survey). Given the fact that large amount of data is available in operators' network, the actual *reactive* cellular network paradigm relying on base station-centric design with *dumb* terminals, should clearly exploit this information to move towards *proactive* context-aware user-centric networks. In other words, big data can enable many solutions for wireless network optimisation such as aggregation of external data sources at the cache-enabled base stations, by exploiting public data from social networks like Facebook and Twitter, localisation info, user velocity, local events, etc. Hence, having caching/computing capabilities at the edge of network and relying on predictability of human behaviour, unleashes further gains in network resources by employing predictive resource management methods and moving strategic information/contents to the edge of network

Based on these motivations, this article explores the potential of big data in wireless cellular networks from a proactive caching perspective, for further improvements in users' experience and backhaul offloading. Indeed, the estimation of content popularity matrix for caching at base stations is a highly challenging task as one has to track spatio-temporal behaviour of users under high data sparsity, large amount of users and content catalog. In order to tackle this, we exploit a big data enabled platform which parallelizes the computation of content popularity via machine learning tools and cache contents at the base stations. As a real world case study, we collect/analyze large amount of data from a telecom operator in Turkey, one of the main mobile operator serving more than 16.2 million of active subscribers. After collecting/analyzing these traces in hours of time interval from several base stations under regulation and privacy concerns, we then study various caching scenarios to assess performance gains for 5G wireless networks.

A. Prior Work and Our Contribution

The potential of big data in mobile computing has been investigated recently in various works such as in [3]. Caching at the edge of network has also received significant attentions as evidenced in [4]–[9]. Compared to these works, our contribution is to explore the potential of big data phenomena in cache-enabled 5G wireless network, by using statistical tools available in machine learning. Supported by a large-scale real-world case study, this is perhaps the first attempt on this direction and shows great potential of big data for cache-enabled 5G wireless networks.

The rest of this paper is organized as follow. In Section II, we describe our network model for proactive caching. In Section III, a practical case study on a big data platform is presented for content popularity estimation, where data extraction process and characterization of user's traffic are given. In Section IV, we show the performance of proactive caching via numerical studies and discuss our results accordingly. Finally, we conclude in Section V and provide future directions.

II. Network Model

Let us consider a network formed by a set of M small base stations (SBSs) denoted by $\mathcal{M} = \{1, \ldots, M\}$ and a set of N user terminals (UTs) denoted by $\mathcal{N} = \{1, \ldots, N\}$. In order to provide broadband Internet connection to users, we assume that each SBS has a wired backhaul link of capacity C_m Mbyte/s and wireless link with total capacity of C'_m Mbyte/s. From the motivation that SBSs are densely deployed, we consider that the backhaul link capacity is limited, thus $C_m < C'_m$. A content library that UTs demand contents (i.e., videos, music, files, news, etc.) is defined as $\mathcal{F} = \{1, \ldots, F\}$. In this library, each content f has a size of $L(f)$ Mbyte with $L_{\text{min}} < L(f) < L_{\text{max}}$ and a bitrate requirement of $B(f)$ Mbyte/s with $B_{\text{min}} < B(f) < B_{\text{max}}$. In fact, the content demand of users follow a Zipf-like distribution $P_{\mathcal{F}}(f), \forall f \in \mathcal{F}$ given as [10]:

$$P_{\mathcal{F}}(f) = \frac{\Omega}{f^{\alpha}} \tag{1}$$

where $\Omega = \left(\sum_{i=1}^{F} \frac{1}{i^{\alpha}} \right)^{-1}$ and the parameter α characterizes the steepness of the distribution. Higher values of α corresponds to steeper distributions which means that a small subset of contents are highly requested, and lower values correspond to more uniform distributions with almost equal probabilities. Note that the value of α depends on users' behaviour and SBSs deployment scenarios (i.e., home, enterprise, urban and rural environments), and its experimental values will be shown in the subsequent sections. Note also that such power laws are used to characterize real-world phenomena, for example the traffic behaviour of cellular devices [11] and popularity of files in the web-proxies [10]. Knowing that the demand follows such a law, let $\mathbf{P}^m(t) \in \mathbb{R}^{N \times F}$ describe the content popularity matrix of the m-th SBS at time t where each coefficient $P^m_{n,f}(t)$ represents the probability of requesting the content f by user n at time t.

In this network model, each SBS has a limited storage capacity of S_m thus caches a subset of contents from the library $\mathcal{F}$. As mentioned before, we aim to avoid the bottlenecks during the delivery caused by the limited-backhaul. In other words, we are interested to characterize the average backhaul load and user's average request satisfaction which are defined hereafter. During the time-interval of T seconds, suppose that D number of contents are requested from the catalog $\mathcal{F}$, denoted by the set $\mathcal{D} = \{1, ..., D\}$. A request $d \in \mathcal{D}$ is immediately served and is called *satisfied* if the bitrate of the requested content $B(f_d)$ is equal or lower than the rate of the delivery, that is

$$\frac{L(f_d)}{\tau'(f_d) - \tau(f_d)} \geq B(f_d) \tag{2}$$

where $\tau(f_d)$ and $\tau'(f_d)$ represent the arrival and end time of delivery for the request f_d respectively. Then, the average *request satisfaction* ratio is expressed as:

$$\eta(\mathcal{D}) = \frac{1}{D} \sum_{d \in \mathcal{D}} \mathbb{1} \left\{ \frac{L(f_d)}{\tau'(f_d) - \tau(f_d)} \geq B(f_d) \right\} \tag{3}$$

where $\mathbb{1}\{...\}$ is the indicator function which takes 1 if the statement holds and 0 otherwise. Now, suppose that the instantaneous backhaul rate at time t for the request d is given by $R_d(t) \leq C_m$ Mbyte/s, $\forall m \in \mathcal{M}$. Then, the average backhaul load is defined as:

$$\rho(\mathcal{D}) = \frac{1}{D} \sum_{d \in \mathcal{D}} \frac{1}{L(f_d)} \sum_{t=\tau(f_d)}^{\tau'(f_d)} R_d(t). \tag{4}$$

In order to minimize the access delays to the contents in such a network, especially during the peak hours, we prefetch the strategic contents at the SBSs, aiming to obtain higher satisfaction ratio and less backhaul load. To show this, let us define the cache decision matrix of SBSs as $\mathbf{X}(t) \in \{0, 1\}^{M \times F}$, where the entry $x_{m,f}(t)$ takes 0 if the f-th content is not cached at m-th SBS at time t, and 1 otherwise. Therefore, one can formally describe the backhaul offloading problem under target request satisfaction and capacity constraints as follows:

$$\begin{aligned}
\underset{\mathbf{X}(t), \mathbf{P}^m(t)}{\text{minimize}} \quad & \rho(\mathcal{D}) && (5)\\
\text{subject to} \quad & L_{\text{min}} \leq L(f_d) \leq L_{\text{max}}, && \forall d \in \mathcal{D},\\
& B_{\text{min}} \leq B(f_d) \leq B_{\text{max}}, && \forall d \in \mathcal{D},\\
& R_d(t) \leq C_m, && \forall t, \forall d \in \mathcal{D}, \forall m \in \mathcal{M},\\
& R'_d(t) \leq C'_m, && \forall t, \forall d \in \mathcal{D}, \forall m \in \mathcal{M},\\
& \sum_{f \in \mathcal{F}} L(f) x_{m,f}(t) \leq S_m, && \forall t, \forall m \in \mathcal{M},\\
& \sum_{n \in \mathcal{N}} \sum_{f \in \mathcal{F}} P^m_{n,f}(t) = 1, && \forall t, \forall m \in \mathcal{M},\\
& x_{m,f}(t) \in \{0, 1\}, && \forall t, \forall f \in \mathcal{F}, \forall m \in \mathcal{M},\\
& \eta_{\text{min}} \leq \eta(\mathcal{D})
\end{aligned}$$

where $R'_d(t)$ Mbyte/s represents the instantaneous rate of wireless link for request d and the parameter η_{min} describes

the minimum target satisfaction ratio. In order to deal with this problem, a joint optimization of cache decision matrix $\mathbf{X}(t)$ and content popularity matrix estimation $\mathbf{P}^m(t)$ is required. However, this is very challenging due to the limited backhaul, wireless link capacities and finite storage of SBSs [6], [12]. Furthermore, large number of users with unknown ratings and big library size have to be considered while dealing such non-tractability. One way to tackle this problem is to enable SBSs and/or their central entity to track, identify, analyse and predict the sparse content popularity/rating matrix $\mathbf{P}^m(t)$.

In this regard, we suppose that the cache placement is done during peak-off hours, therefore $\mathbf{X}(t)$ is represented by a cache decision matrix $\mathbf{X}$ which remains static during in peak hours. Moreover, we consider that all SBSs have an identical and stationary content popularity matrix over T time slots, thus we represent $\mathbf{P}^m(t)$ by $\mathbf{P}$. Indeed, if we have sufficient amount of users' ratings, we can approximately design a k-rank popularity matrix $\mathbf{P} \approx \mathbf{N}^T\mathbf{F}$, where $\mathbf{N} \in \mathbb{R}^{k\times N}$ and $\mathbf{F} \in \mathbb{R}^{k\times F}$ are the factor matrices obtained from joint learning, that minimizes the following cost function:

$$\underset{(i,j)\in\mathbf{P}}{\text{minimize}} \sum_{(i,j)\in\mathbf{P}} \Big(\mathbf{n}_i^T\mathbf{f}_j - P_{ij}\Big)^2 + \mu\Big(||\mathbf{N}||_F^2 + ||\mathbf{F}||_F^2\Big) \quad (6)$$

where the vectors $\mathbf{n}_i$ and $\mathbf{f}_j$ describe the i-th and j-th columns of $\mathbf{N}$ and $\mathbf{F}$ matrices respectively, $||.||_F^2$ is the Frobenius norm, and above summation is done over the user/content rating pairs (i,j) in the training set. The role of parameter μ is to provide a balance between fitting training data and regularization.

In practice, in order to estimate $\mathbf{P}$ in (6), we use a *big data platform* of the aforementioned network operator. Based on this estimation, we then store contents at the *cache-enabled base stations* whose cache decisions are represented by $\mathbf{X}$. By doing so, minimization problem of backhaul offloading in (5) is attained. The network model consists of big data platform and cache-enabled SBSs is illustrated in Fig. 1. The following section is dedicated to details of our *big data platform* and characterisation of users' traffic pattern.

III. Big Data Platform

The big data platform used in this work analysis user's data traffic and runs in the operator's core network. In other words, the purpose of this platform is to extract useful information for proactive caching decisions and store users' traffic data. In brief, the operator network under investigation covers more than 10 regional core areas in several districts in Turkey. The average total traffic consists of approximately over 20 billion packets in the downlink direction and over 15 billion packets in the uplink direction over all territorial areas. Consequently, the core network of the mobile operator is loaded by approximately over 80 TByte of total data in uplink and downlink daily. Also, an exponential increase of data usage is observed in this mobile operator. For instance, in 2012, the daily uplink and downlink traffic was approximately over 7 TByte.

In our work, we collect the streaming traces in a server with high speed link of 200 Mbit/sec at peak hours. This is done by

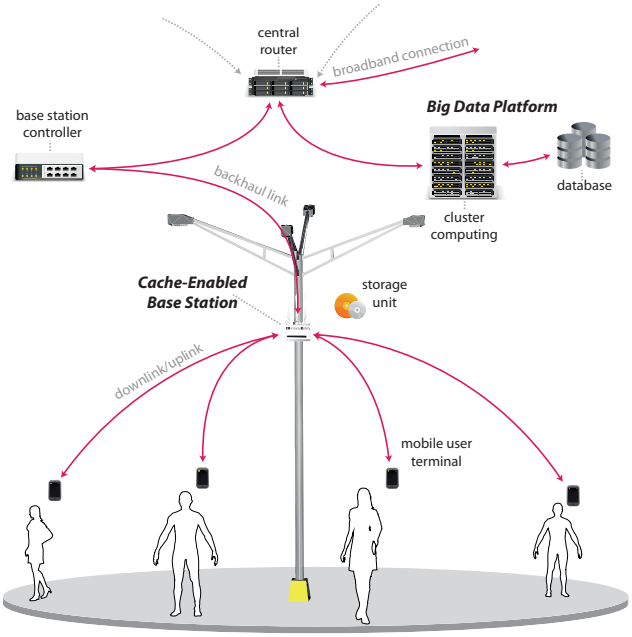


Figure 1: Illustration of the proactive network model. The *cache-enabled base stations* with limited storage unit satisfy their users, based on the strategic contents predicted on the *big data platform*.

initializing a mirroring task via real-word Gn interface data.[1] After the mirroring stage, network traffic is then transferred into the server on the platform. For our practical case study, we have collected traffic of approximately 7 hours starting from 12 PM to 7 PM on Saturday 21'st of March 2015, and is processed on Hadoop platform.

A. Hadoop platform

Hadoop is one of the key big data framework that supports massive volumes of transactional data at the lowest level of granularity. It supports implementation and execution of distributed applications on large clusters. A computational model named Map-Reduce enables execution of an application in parallel, by dividing it into many small fragments of jobs on multiple nodes. The storage module, called Hadoop Distributed File System (HDFS), is in charge of handling the data storage across the clusters.

In order to test the precision of proposed mechanism in the operator's network, we use a platform based on Cloudera's Distribution Including Apache Hadoop (CDH4) [13] version on four nodes including one cluster name node, with computations powers corresponding to each node with INTEL Xeon CPU E5-2670 running @2.6 GHz, 32 Core CPUs, 132 GByte RAM, 20 TByte hard disk. In the following section, we describe the data extraction process on this platform.

[1]Gn is an interface between Serving GPRS Support Node (SGCN) and Gateway GPRS Support Node (GGSN). Network packets sent from a user terminal to the packet data network (PDN), e.g. internet, pass through SGCN and GGSN where GPRS Tunneling Protocol (GTP) constitutes the main protocol in network packets flowing through Gn interface.

B. Data Extraction Process

First, we parse raw data using Wireshark command line utility *tshark* in order to extract relevant fields of CELL-ID (or service area code (SAC) in our case, used to uniquely identify a *service area* within a *location area*[2]), LAC, Hypertext Transfer Protocol (HTTP)-request-uniform resource identifier (URI), tunnel endpoint identifier (TEID)[3] and TEID-DATA for data and control plane packets respectively, and FRAME TIME indicating arrival time of packets. The HTTP Request-URI is a Uniform Resource Identifier that identifies the resource upon which to apply the request. The *control* packets contain information required for future data packets. In particular, it contains cell identification ID (CELL-ID), LAC and TEID-DATA fields. The *data* packets contain HTTP-URI and TEID fields.

The next step consists of transferring data with these relevant fields into HDFS for further analysis. This allows to process data, and apply different data analytics over header information of both control and data planes using high level query languages such as Hive Query language (QL) and Pig Latin, where the concept of Map-Reduce is inherently embedded. For instance, in order to calculate the HTTP Request-URIs at given location, the HTTP-URI can be merged with CELL-ID-LAC fields over the same TEID and TEID-DATA fields for data and control packets respectively. In our analysis, due to the limitations on collected number of rows for HTTP-URI fields matching with CELL-ID-LAC fields, we have proceeded with HTTP Request-URIs and TEID mappings.

In order to get the final table called *traces-table* with fields of SIZE, HTTP Request-URIs, FRAME TIME and TEID, we first construct a temporary table called *traces-table-temp* using Hive QL, having same fields as *traces-table* except SIZE. Then, we calculate the sizes of each HTTP Request-URI request using a separate URI-*size calculator* program that uses HTTPClient API. In the end, the *traces-table* table has approximately over 420.000 of 4 millions HTTP Request-URIs with SIZE field returned as not zero or null due to unavailability of HTTP response for some requests. Note that different HTTP Request-URIs can appear in a given session with a specific TEID. Each TEID corresponds to a specific user. Also, each user can have different TEIDs with different HTTP Request-URIs. We summarize the steps of data extraction process in Fig. 2.

C. Traffic Characteristics

Using the final traces, we illustrate the global content popularity distribution (namely HTTP-URI popularity distribution) in Fig. 3a, in which the contents are ranked in decreased order based on their popularity. This figure shows that the content popularity behaviour follows a Zipf law with shape parameter $\alpha = 1.36$. Moreover, we illustrate the cumulative size of ranked contents in Fig. 3b. In this figure, the cumulative size up to 41-th most-popular contents has 0.1 GByte of size, whereas a huge increase appears afterwards, meaning that majority of the demanded contents in our traces have short sizes while contents with big sizes are relatively less demanded.

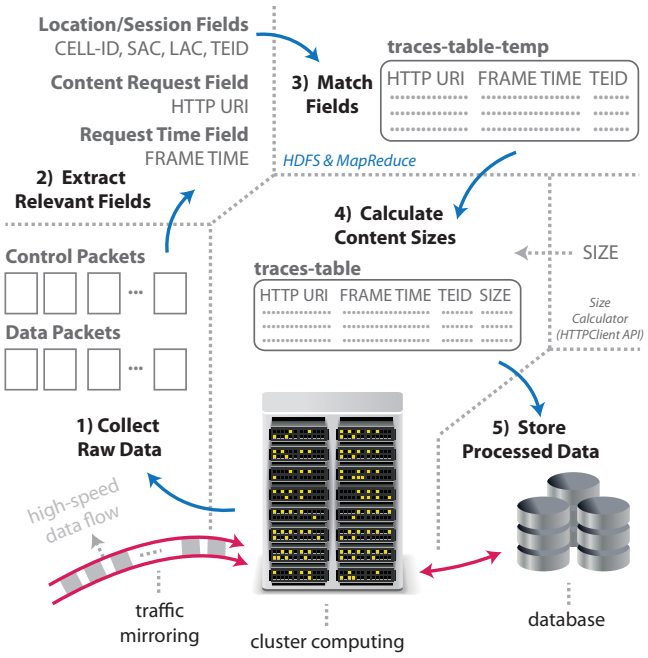


Figure 2: An illustration of the data extraction process on the big data platform.

We would like to note that a detailed characterization of traffic patterns for caching is under investigation. For a similar study in terms of characterization of traffic can be found in [14]. Different than [14], we focus on the traffic characterization in a large regional area for proactive caching (i.e., content popularity distribution, cumulative size distribution). In the following section, we simulate the scenario of cache-enabled SBSs based on available information in *traces-table*.

IV. Numerical Results and Discussions

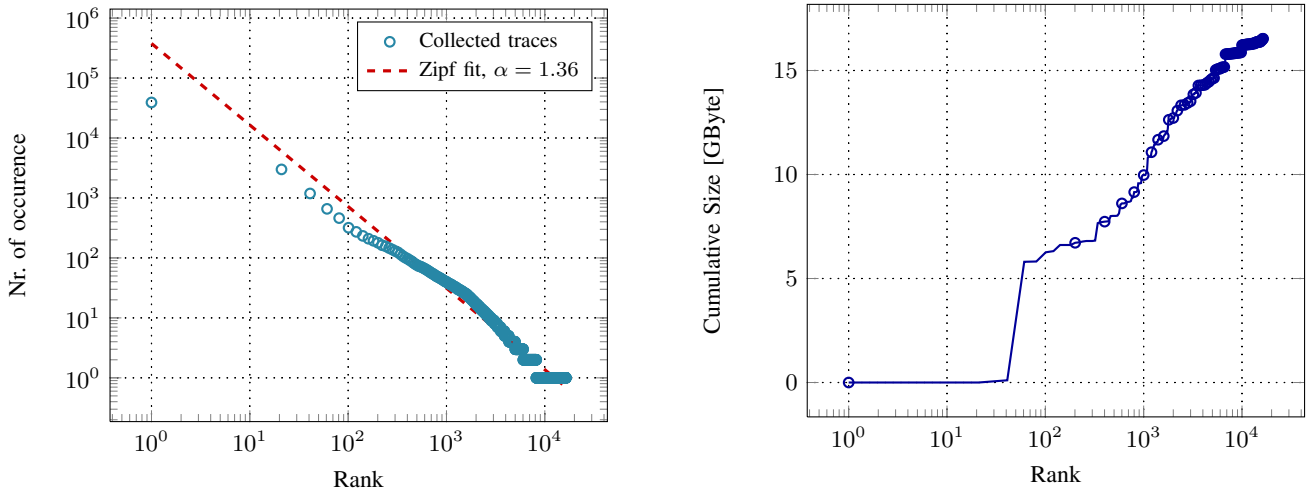
In the numerical setup, the storage capacities, backhaul and wireless link capacities of SBSs are set to identical values for ease of analysis and revealing of caching gains. The values of parameters in the numerical setup is summarized in Table I.

In the simulations, D number of requests from *traces-table* are considered, spanning over 6 hours 47 minutes of time interval. More precisely, the arrival times (FRAME TIME), requested contents (HTTP-URI) and content sizes (SIZE) are taken from this table, and associated to M base stations pseudo-randomly. The following two methods are used for comparison:

- *Ground Truth*: Content popularity matrix $\mathbf{P}$ is constructed by considering all available information in *traces-table*.

[2]The service area represented by SAC is an area of base station(s), and corresponds to a location area which is uniquely represented by location area code (LAC). In general, tens or hundreds of base stations operates in a given location area.

[3]A TEID uniquely represents a tunnel endpoint on receiving end of the GTP tunnel. A local TEID value is assigned at receiving end of a GTP tunnel in order to send messages through the tunnel.



(a) Global distribution of content popularity.

(b) Cumulative distribution of content sizes.

Figure 3: Content popularity distribution gathered from real traces.

Table I: List of simulation parameters.

Parameter	Description	Value
T	Time slots	6 hours 47 minutes
D	Number of requests	422529
F	Number of contents	16419
M	Number of small cells	16
$L_{\min}$	Min. size of a content	1 Byte
$L_{\max}$	Max. size of a content	6.024 GByte
$B(f)$	Bitrate of content f	4 Mbyte/s
$\sum_m C_m$	Total backhaul link capacity	3.8 Mbyte/s
$\sum_m \sum_n C'_m$	Total wireless link capacity	120 Mbyte/s

This matrix has 6.42% of rating density.

- *Collaborative Filtering*: For estimation of $\mathbf{P}$, 10% of ratings in *traces-table* are chosen uniformly at random and given to collaborative filtering (CF) for training. Then, the remaining ratings are predicted via regularized singular value decomposition (SVD) [15].

Relying on knowledge of content popularity from these methods, the most-popular contents are proactively cached at the SBSs until reaching their storage capacity. The simulation is then carried out starting from the first content delivery at $t = 0$ until the completion of last request.

A. Users' Request Satisfaction

The impact of storage size on the users' request satisfaction is shown in Fig. 4a. Note that 100% of storage size in the figure corresponds to caching of entire catalog (17.7 GByte in our case), and 0% corresponds to no caching. From the figure, we observe that the satisfaction is monotonically increasing as the storage size increases, whereas a performance gap between the ground truth and CF is experienced until 87% of storage size, mainly due to the estimation errors. For example, the ground truth and CF achieve 92% and 69% of satisfaction respectively when considering 40% of storage size.

B. Backhaul Usage

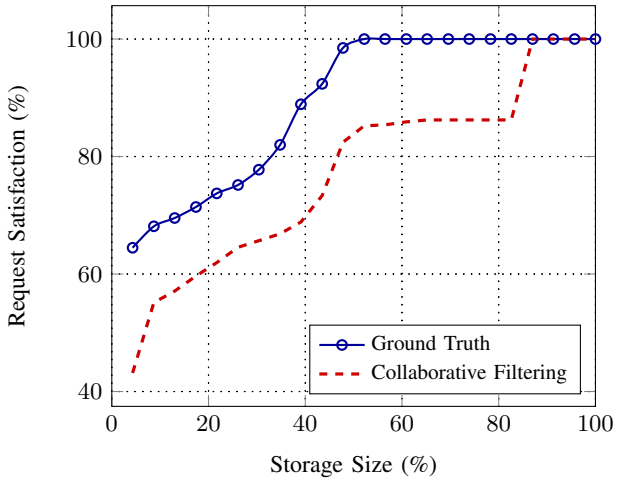
The impact of storage size on the backhaul load/usage is shown in Fig. 4b. In this figure, it is clear that increment in storage size yields higher offloading gains (namely less backhaul usage). For example, we see that having 87% of storage size reduces 98% of backhaul usage. On the other hand, the performance of ground truth is higher than the CF approach in intermediate values of storage size. This is due to the fact that popularity-based cache placement with non-identical content sizes in catalog results in high backhaul usage, especially when relatively popular (but big sized) contents are not cached (see Fig. 3b). This clearly points out the importance of size distribution in cache strategies, in addition to consideration of content popularity distribution.

C. Impact of Rating Density

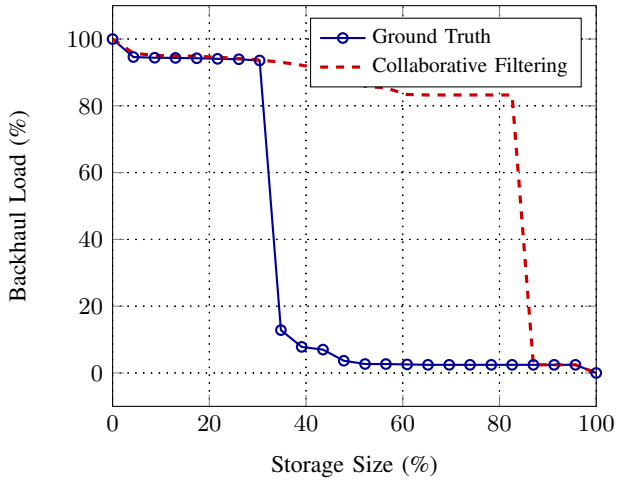
So far, we have compared the performance of these approaches under the setting of 10% of rating density in CF. Indeed, the rating density is crucial in CF as non-sufficient amount of ratings might lead to high estimation errors (sometimes called cold-start problem). To show this, the impact of rating density on the root-mean-square error (RMSE) is shown in Fig. 5, whereas the error is defined as the root-mean-square difference of users' request satisfaction in both methods over all possible storage sizes. As confirmed in Fig. 5, the performance of CF highly depends on the availability of ratings, where increase of ratings results in less errors thus yielding higher performance.

V. Conclusions

We have studied a proactive caching scheme for 5G mobile cellular networks where huge amount of available data is exploited for content popularity estimation via machine learning



(a) Impact of storage size on the request satisfaction.



(b) Impact of storage size on the backhaul usage.

Figure 4: Numerical results for practice caching at the base stations.

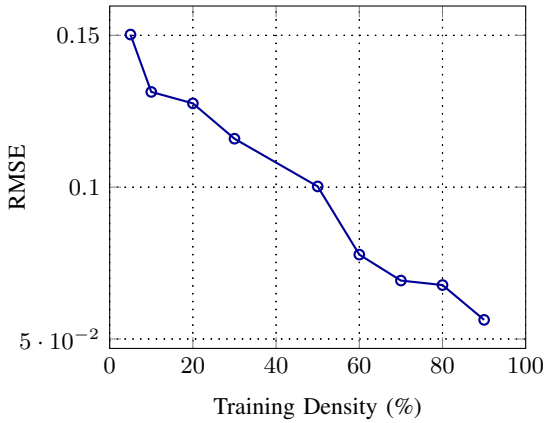


Figure 5: Impact of training density on the RMSE.

tools. In particular, we have demonstrated the collection/extraction process of our real traces on a big data platform, then make use of machine learning tools for content popularity estimation. Subsequently, caching at the base stations is carried out through numerical studies, to show the benefits of our approach for 5G wireless networks. We led to a conclusion that several gains in terms of users' satisfaction and backhaul offloading are possible, and depend on available rating density and storage size.

One possible direction of this work is to provide a more detailed characterization of the traffic which can reflect various spatio-temporal content access patterns. In this regard, design of novel machine learning tools are also needed so that cache placement at the base stations can be applied more efficiently. Design of deterministic/randomized cache placement algorithms is also of high interest and should not purely rely on content popularity.

References

[1] J. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. Soong, and J. Zhang, "What will 5G be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[2] H. Hu, Y. Wen, T.-S. Chua, and X. Li, "Toward scalable systems for big data analytics: A technology tutorial," *IEEE Access*, vol. 2, pp. 652–687, 2014.

[3] J. K. Laurila, D. Gatica-Perez, I. Aad, B. J., O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, and M. Miettinen, "The Mobile Data Challenge: Big Data for Mobile Computing Research," in *Pervasive Computing*, 2012.

[4] E. Baştuğ, M. Bennis, and M. Debbah, "Living on the Edge: The role of proactive caching in 5G wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82 – 89, August 2014.

[5] E. Baştuğ, M. Bennis, M. Kountouris, and M. Debbah, "Cache-enabled small cell networks: Modeling and tradeoffs," *EURASIP Journal on Wireless Communications and Networking*, no. 1, p. 41, February 2015.

[6] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 142–149, 2013.

[7] D. Liu and C. Yang, "Energy efficiency of downlink networks with caching at base stations," *arXiv preprint arXiv:1505.06615*, 2015.

[8] D. Malak and M. Al-Shalash, "Optimal caching for device-to-device content distribution in 5g networks," in *Globecom Workshops (GC Wkshps), 2014*, December 2014, pp. 863–868.

[9] V. S. Varma and T. Q. S. Quek, "Congestion games in caching enabled heterogeneous cellular networks," in *14th International IFIP TC6 Networking Conference, Networking*, Toulouse, France, May 2015.

[10] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *IEEE Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99)*, vol. 1. IEEE, 1999, pp. 126–134.

[11] M. Z. Shafiq, L. Ji, A. X. Liu, and J. Wang, "Characterizing and modeling internet traffic dynamics of cellular devices," in *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*. ACM, 2011, pp. 305–316.

[12] E. Baştuğ, J.-L. Guénégo, and M. Debbah, "Proactive small cell networks," in *20th International Conference on Telecommunications (ICT'13)*, Casablanca, Morocco, May 2013.

[13] "Cloudera," http://goo.gl/dfkWib, 2015, [Online; accessed 02-April-2015].

[14] E. Mucelli Rezende Oliveira, A. Carneiro Viana, K. P. Naveen, and C. Sarraute, "Measurement-driven mobile data traffic modeling in a large metropolitan area," INRIA, Research Report RR-8613, October 2014.

[15] J. Lee, M. Sun, and G. Lebanon, "A comparative study of collaborative filtering algorithms," *[Online] arXiv: 1205.3193*, 2012.