# Performance Gain of Precaching at Users in Small Cell Networks

Binqiang Chen and Chenyang Yang

Beihang University, Beijing, China
Email: chenbq@buaa.edu.cn, cyyang@buaa.edu.cn

***Abstract*—The throughput of small cell networks (SCNs) is limited by intercell interference. Except numerous techniques of interference coordination, simply caching several popular files at each user provides an alternative approach to reduce interference by pre-offloading. This becomes possible nowadays due to the facts that the cost of memory devices is reducing quickly and content delivery is gradually dominating the traffic load. In this paper, we quantify the performance gain introduced by precaching at users. We derive the average throughput for non-coordinated downlink SCN with precaching, and compare with the SCN without caching and with existing prefetching policy. We use simulation to validate the analytical results and further evaluate the average download time of different policies. Our results show that the caching gain is remarkable when the traffic load is high and the popularity distribution is far from uniform.**

***Index Terms*—Caching, interference, small cell networks**

## I. Introduction

Small cell networks (SCNs) is promising to support high throughput for 5G cellular networks, where one of the limiting factors is intercell interference (ICI) [1]. While various interference coordination techniques are available in the literature (e.g., [2]), in order to meet the explosive growth of traffic demands in a cost effective way, alternative approaches need to be developed by rethinking the usage model of the networks.

Recently, it has been observed that a large portion of the traffic is generated by many duplicate downloads of a few popular contents. This naturally calls for caching inside the networks. In cellular networks, caching popular files at the base stations (BSs) or even at users can reduce the backhaul cost and access latency or boost the throughput [3–5]. Noticing that backhual is becoming a bottleneck for SCNs meanwhile storage capacity grows rapidly with relative low cost, the authors in [3] suggested to cache at small BSs or at mobile users with device to device (D2D) transmissions to improve the performance of networks.

In fact, caching content at users has long been applied as an important technique for improving the quality of user experience (QoE) [6]. Prefetching contents to users before the users request the files can reduce the user perceived delay, especially in poor and intermittently connected wireless networks [6, 7]. Caching has also been considered a technique to offload the traffic in cellular networks to Wi-Fi by prefetching popular contents in memories at users when Wi-Fi is available [8].

The success of prefetching largely depends on the accuracy of prediction on the interest of each specific user. Moreover, the contents need to be opportunistically pushed to the cache of a user, because the user may not be in the coverage of Wi-Fi or in good channel condition, or the cellular network may be busy with other on-demand traffic [6, 9]. Existing research in this context focus on answering the questions of which content should be prefetched to which user, and when to push the contents to the users. On the other hand, in cache-enabled D2D [3], different contents are pushed to different users within a cluster. By jointly optimizing clustering and content placement, the network throughput can be increased.

Inspired by the idea of improving network performance and user experience by caching at users, we consider a more aggressive way of caching. Specifically, with the statistic information of the user demands and traffic load in the network, the operators can simply broadcast several popular files to all users in off-peak times rather than only find excess resources for pushing the contents. This is essentially pre-offloading, which has been recognized as an efficient way to improve throughput in content centric networks [10]. We call such a cache placement policy as *precaching*. It is no doubt that caching popular contents at users can definitely bring performance gain, as reported in many recent research efforts, e.g., [3–5]. However, the cache size at a user is small compared to the file catalog size. Moreover, a user may not request the files stored in its cache, and still needs to download the files from the BS. Further considering the complex interaction among content popularity, randomly arrived requests, and interference level, how large throughput gain can be provided by such a simple policy and whether user experience can be improved in general with acceptable cache size remain unclear.

In this paper, we strive to quantify the performance gain from precaching in non-coordinated downlink SCNs. While pre-offloading is not an interference coordination technique, it can naturally mitigate the ICI. To this end, we first derive the average throughput of the SCNs respectively with or without precaching, and compare with the throughput of prefetching. Then, we derive caching gain in throughput for precaching and prefetching over the non-coordinated SCN without caching. We use simulation to validate the analytical results. To reflect

This work was supported by the National High Technology Research and Development Program of China (No. 2014AA01A703) and National Natural Science Foundation of China (No. 61429101).

the user experience, we also simulate the average download time. The results demonstrate that the throughput gain and the download time reduction from precaching are significant, especially when the average arrival rate is high and the cached files are more popular.

## II. System Model

Consider a non-coordinated downlink SCN with $B$ small BSs, each equipped with $M$ antennas. There are multiple single antenna users uniformly located in the cells. For mathematical simplicity, we consider circle cells each with radius $D$. Suppose that a macro BS is co-located with these small BSs in the same area but operates in a different frequency band. Hence, the macro BS and small BSs can transmit simultaneously to different group of users without interference.

We consider static catalog including $N_f$ files that the users may request, where the files are indexed according to popularity, e.g., the 1st file is the most popular file. For notational simplicity, each file is assumed with the same size of $F$ bits. The probability that the $i$th file is requested by a user follows Zipf distribution, which is $P_{N_f}(i) = i^{-\beta}/\sum_{k=1}^{N_f} k^{-\beta}$, where the parameter $\beta$ reflects the popularity of the files [11]. The BSs are connected to the core network with backhaul links, and each user has local cache with capacity of $C_u$ Bytes.

Assume that each user requests one file from the catalog, where the request follows Poisson process with identical average arrival rate $\lambda$ (files/s).

We consider the following simple cache placement policy. Every user caches the most popular $N_u$ files, which occupy $r_c \triangleq N_u F/C_u$ portion of the local cache capacity, $N_u < N_f$. In practice, these files can be proactively downloaded by the operator from the macro BS to the caches at each user via broadcasting during off-peak times according to the statistics of the user demand. In such a way, the pre-offloading procedure for these popular files will cause negligible or even no performance degradation to the performance of the network. When a request of a user truly arrives, if the file requested is in the user's own cache, then the user can directly obtain the file from itself with perfect QoE, i.e., with zero download time. Otherwise, the file will be fetched via backhaul and then transmitted to the user by a small BS closest to it (called *local BS of the user*). We refer to such a policy as *Precaching*.

If a small BS has no request to serve, then the BS will be turned into idle mode immediately to avoid interference.

We consider block fading channels, which remain constant in each block and are independent among blocks. The small-scale fading channel from the $i$th BS to the $j$th user is denoted as $\boldsymbol{h}_{ji}$, and the distance between them is denoted as $l_{ji}$.

## III. Average Throughput and Caching Gain

With precaching, a part of traffic load during peak hours can be pre-offloaded. To quantify the performance gain from precaching, in this section we derive the average throughput of the considered SCN with the simple policy, and compare with a baseline network without caching and another cache-enabled network with prefetching. Then, we analyze the corresponding caching gains over the baseline.

### *A. Average Throughput*

The average throughput of the SCN is defined as the overall number of bits transmitted in the $B$ cells in unit time averaged over small scale channel fading and user location.

To simplify the analysis and capture the essence of the problem, we assume that each BS only serves one user closest to it (called *local user*) each time. The impact of multiple users will be evaluated later in section IV.

In a SCN, when traffic load is very low, many BSs sleep and the ICI is weak. As a result, the average service rate for every user exceeds the data arrival rate at every active BS $\lambda F$ (bits/s), and the active BSs can convey all the requested files to the users [12]. However, when traffic load is high, since many BSs need to transmit simultaneously, the average service rates for the users who are far away from the BSs may be less than the data arrival rate due to ICI.

To derive the average throughput, we classify each small cell, say the $i$th cell, into two areas $\mathcal{A}_i$ and $\mathcal{B}_i$, according to the location of a user. If an active user is located in $\mathcal{A}_i$, its average service rate $\mathbb{E}_{\boldsymbol{h}}\{R_i\} \geq \lambda F$. If an active user is located in $\mathcal{B}_i$, $\mathbb{E}_{\boldsymbol{h}}\{R_i\} < \lambda F$, then the $i$th small BS needs admission control to avoid congestion. Specifically, if a small BS is busy with serving a previous request meanwhile there is one request waiting in its queue, then the newly arrived request will not be accessed to keep the system stable. The sizes of the two areas depend on the interference level, which will be addressed later. Since the users are uniformly located, the average throughput of the SCN can be derived as

$$\bar{C} = \frac{1}{\pi D^2} \sum_{i=1}^{B} \Big( \iint_{\mathcal{A}_i} \lambda F d\mathcal{A}_i + \iint_{\mathcal{B}_i} \mathbb{E}_{\boldsymbol{h}}\{R_i\} d\mathcal{B}_i \Big). \tag{1}$$

Since the role of the macro BS in this work is to broadcast the popular files to the users, and we are only concerned with the performance of the SCN, in the sequel we call a small cell as a cell, and a small BS as a BS for simplicity.

To demonstrate the throughput gain from precaching, we first analyze the average throughput of a baseline SCN, where no files are cached at users and all requests are served by the BSs, which is referred to as *Policy 0*.

*1) Policy 0 (Non-Caching):* Without coordination, a BS simply ignores the ICI and transmits to its local user with maximum ratio transmit (MRT) when a request arrives. The precoding vector at the $i$th BS is $\boldsymbol{w}_i = \sqrt{P}\boldsymbol{h}_{ii}/\|\boldsymbol{h}_{ii}\|$, where $P$ is the transmit power at the BS, and $\|\cdot\|$ stands for Euclidean norm. Then, the signal to interference plus noise ratio (SINR) of the $i$th user served by the $i$th BS can be expressed as

$$\gamma_i = \frac{P}{l_{ii}^{\alpha}(I_i + \sigma^2)} g_{ii}, \tag{2}$$

where $g_{ij} \triangleq \|\boldsymbol{h}_{ij}^H \boldsymbol{w}_j\|^2$, $I_i = \Sigma_{j=1, j\neq i}^{B} l_{ij}^{-\alpha} g_{ij}$ is the power of ICI, $\alpha$ is the attenuation factor, $\sigma^2$ is the variance of white Gaussian noise, and $(\cdot)^H$ denotes conjugate transpose.

Assuming Rayleigh fading with unit variance, then $g_{ij}$ follows Gamma distribution. By using Proposition 9 in [13], the average service rate of the $i$th user can be obtained as

$$\mathbb{E}_{\boldsymbol{h}}\{R_i\} \approx W\log_2(1+\frac{MP}{l_{ii}^{\alpha}(\mathbb{E}_{\boldsymbol{h}}\{I_i\}+\sigma^2)}), \quad (3)$$

where $W$ is the bandwidth, $\mathbb{E}_{\boldsymbol{h}}\{I_i\} = P\Sigma_{j=1,j\neq i}^{B} l_{ij}^{-\alpha}$, and the approximation is accurate when the number of antennas at each BS, $M$, is large.

According to (1), when the file arrival rate $\lambda$ is low such that $\mathbb{E}_{\boldsymbol{h}}\{R_i\} \geq \lambda F$ for all active users, i.e., $\mathcal{B}_i = 0$, the average throughput is equal to the average data arrival rate in all cells, $\bar{C}_0 = \lambda F B$.

In general cases, $\mathcal{B}_i \neq 0$. With high traffic load, many BSs are active and the average throughput depends on the ICI power. To simplify the analysis, we approximate $\mathbb{E}_{\boldsymbol{h}}\{I_i\}$ in (3) as follows. For a user located in the shadow area in Fig. 1, the average ICI power can be approximated by considering the nearest six BSs around the user as

$$\mathbb{E}_{\boldsymbol{h}}\{I_i\} \approx \bar{I}_0$$
$$= \frac{6}{A}\int_{D}^{3D}\frac{P}{l^{\alpha}}\frac{2\pi l}{6}dl = \frac{3P}{2(\alpha-2)(D^{-\alpha}-9(3D)^{-\alpha})},$$

where $A = 4\pi D^2/3$ is the area of the shadow area. In fact, we can consider all interference in the network, which however yields much complex expression that is less useful for further analysis. Considering that the interference generated by the BSs far away from the user has little contribution to the average ICI power, the approximation is accurate as illustrated later via simulations.

Then, from (3) the average service rate can be further approximated as

$$\mathbb{E}_{\boldsymbol{h}}\{R_i\} \approx W\log_2(1+\frac{u_0}{l_{ii}^{\alpha}}), \quad (4)$$

where $u_0 = MP/(\bar{I}_0+\sigma^2)$.

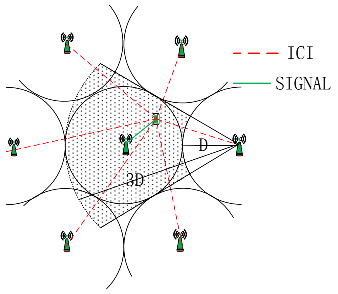

Fig. 1. Approximated average ICI

To derive the average throughput for the high traffic load scenario, in what follows, we show how to divide the $i$th cell into the two areas $\mathcal{A}_i$ and $\mathcal{B}_i$. As shown in (4), $\mathbb{E}_{\boldsymbol{h}}\{R_i\}$ is a decreasing function of $l_{ii}$. Consequently, there must exist a distance $D_0$ $(0 \leq D_0 \leq D)$, such that $\mathbb{E}_{\boldsymbol{h}}\{R_i\} \geq \lambda F$ for $\forall l_{ii} \leq D_0$. By setting $\mathbb{E}_{\boldsymbol{h}}\{R_i\} = \lambda F$, $D_0$ can be obtained as

$$D_0 = \min\{(\frac{u_0}{(2^{\lambda F/W}-1)})^{1/\alpha}, D\}, \quad (5)$$

which is referred to as the *distance threshold for policy* 0. For the user served by the $i$th BS, if $l_{ii} \leq D_0$, then the user is located in $\mathcal{A}_i$. Otherwise, the user is located in $\mathcal{B}_i$.

Then, from (1), (4) and (5), the average throughput of the baseline network can be derived as

$$\begin{aligned}\bar{C}_0 &\approx \frac{B}{\pi D^2}(\lambda F\pi D_0^2 + \int_{D_0}^{D} W\log_2(1+\frac{u_0}{l^{\alpha}})2\pi l dl)\\ &\approx (\frac{D_0}{D})^2 B\lambda F + \frac{\alpha BW}{2\ln 2}(1-(\frac{D_0}{D})^2) + BW(\log_2(1+\frac{u_0}{D^{\alpha}})\\ &- (\frac{D_0}{D})^2\log_2(1+\frac{u_0}{D_0^{\alpha}}))\\ &\triangleq f(D_0,\lambda), \end{aligned} \quad (6)$$

where the second approximation is accurate when the average cell edge SINR $u_0 D^{-\alpha}$ is high (note that $MPD^{-\alpha}$ reflects the average receive signal for a user located exactly at the cell-edge, and $\bar{I}_0+\sigma^2$ is the average interference plus noise).

*2) Policy 1 (Precaching):* Since with this policy every user caches the most popular $N_u$ files, the probability that the requested file of a user is cached at the user, i.e., the *cache hit ratio*, is $P_1^u = \sum_{j=1}^{N_u} j^{-\beta}/\sum_{i=1}^{N_f} i^{-\beta}$ [11].

If the requested file of a user is in its local cache, the user is called a *cache hit user*, and the event is named a *cache hit event*. Otherwise, the user is call a *cache miss user*, whose file needs to be served by its local BS with MRT, and the event is named a *cache miss event*. The cache hit users enjoy perfect QoE and are naturally isolated from the ICI. On the other hand, the cache hit users do not generate interference to the on-going traffic of the cache miss users. Therefore, the average throughput of the network contains two parts: $\bar{C}_1^u$ contributed by the caches at the users and $\bar{C}_1^b$ contributed by the BSs with reduced ICI.

Since cache hit and miss events occur independently and each user has identical cache hit ratio $P_1^u$, the cache hit events of all cache hit users follow Bernoulli distribution with probability $P_1^u$. Then, the average number of the cache hit users is $P_1^u\lambda B$, where $\lambda B$ is the average number of users arrived in all the $B$ cells. Sequentially, the average throughput contributed by the caches can be easily obtained as

$$\bar{C}_1^u = \lambda F P_1^u B. \quad (7)$$

By precaching during off-peak times, $P_1^u\lambda B$ of the arrived requests are "served" immediately by the local caches, meanwhile these requests do not generate interference to other on-going traffic. For the remaining $\lambda_1 = (1-P_1^u)\lambda$ file requests to be served by BSs, we can use similar way to derive the average throughput as for *policy* 0. Specifically, we can derive a *distance threshold for precaching* as

$$D_1 = \min((\frac{u_0}{(2^{\lambda_1 F/W}-1)})^{1/\alpha}, D). \quad (8)$$

Again, when the cell edge SINR $u_0 D^{-\alpha}$ is high, the average throughput contributed by the BSs after the pre-offloading can be derived as $\bar{C}_1^b \approx f(D_1,\lambda_1)$, where the function $f(\cdot)$ is defined in (6). Further considering (7), the average throughput of the network is

$$\bar{C}_1 = \bar{C}_1^u + \bar{C}_1^b. \quad (9)$$

*3) Policy 2 (Prefetching):* Existing prefetching policies pre-downloads the files to the users who have installed corresponding application software by predicting their interest [7–9]. To avoid degrading the performance of a busy cellular network or exhausting the energy of a mobile terminal, the contents are pushed to the cache of a user only when a cellular network has excess radio resource and the user is with good channel condition [9], or a Wi-Fi is in the coverage [8]. As a result, the files can only be pre-downloaded to few users.

To model different interests of the users and the uncertainty of the prediction in each user's interest, we assume that each user may be interested in a file from a set with size $N_s$, which is a subset of the static catalog, as in [9]. To be specific, each subset is obtained from a realization of Zipf distributed random variables over all the $N_f$ files, where the most popular $N_s$ files are selected. Therefore, the subsets for different users may differ. The probability that the $i$th file in a subset is requested by a user follows Zipf distribution, i.e., $P_{N_s}(i) = i^{-\beta_s}/\sum_{k=1}^{N_s} k^{-\beta_s}$, where $\beta_s$ is the parameter reflecting the uncertainty in prediction. Notice that $\beta_s$ is different from $\beta$, which reflects the skewness of the interests of all users in the SCN [9].

Suppose that only $r_u$ percentage of the users participate the prefeching, each user is cached with the most popular $N_u^s$ files in the subset for the user, $0 \leq r_u \leq 1$. Then, prefetching can be analyzed as a special case of precaching. It is not hard to show that the cache hit ratio of prefetching is $P_2^u = r_u \sum_{j=1}^{N_u^s} j^{-\beta_s}/\sum_{i=1}^{N_s} i^{-\beta_s}$, and the average throughput contributed by prefetching at users can be easily obtained as

$$\bar{C}_2^u = \lambda F P_2^u B. \tag{10}$$

The remaining requests need to be served by the BSs is $\lambda_2 = (1 - P_2^u)\lambda$. With a similar method for *policy* 0, we can derive a *distance threshold for prefetching* as

$$D_2 = \min\left(\left(\frac{u_0}{(2^{\lambda_2 F/W} - 1)}\right)^{1/\alpha}, D\right). \tag{11}$$

Again, when the cell edge SINR $u_0 D^{-\alpha}$ is high, the average throughput contributed by the BSs can be derived as $\bar{C}_2^b \approx f(D_2, \lambda_2)$. Further considering (10), the average throughput of the network with prefeching is

$$\bar{C}_2 = \bar{C}_2^u + \bar{C}_2^b. \tag{12}$$

## *B. Caching Gain*

To reflect the throughput gain brought by caching at the users, we define caching gain as

$$G_{ci} \triangleq \bar{C}_i - \bar{C}_0, \tag{13}$$

where $\bar{C}_i$ are the average throughput of the SCN with policy $i, i \in \{1, 2\}$, respectively.

In practice, heavy traffic load scenarios are more interesting, since for small values of $\lambda$ even the baseline network can transmit all requested files reliably. Therefore, we focus on the caching gain when the file arrival rate $\lambda$ is large.

*1) Policy 1 (Precaching):* By substituting (6) and (9) into (13), the caching gain can be derived as

$$\begin{aligned}
G_{c1} &= \bar{C}_1 - \bar{C}_0 = (P_1^u \lambda F B \\
&- \frac{D_1^2}{D^2} B(P_1^u \lambda F + W \log_2(1 + \frac{u_0}{D_0^\alpha}) - W \log_2(1 + \frac{u_0}{D_1^\alpha})) \\
&+ \frac{\Delta D_1^2}{D^2}(\lambda F B - BW \log_2(1 + \frac{MP}{(\bar{I}_0 + \sigma^2) D_0^\alpha}) - BW \frac{\alpha}{\ln 2})) \\
&\triangleq g(P_1^u, D_1),
\end{aligned} \tag{14}$$

where $\Delta D_1^2 = D_1^2 - D_0^2$.

When the number of cached files at each user $N_u$ is small and the traffic load is heavy, most BSs are busy. Specifically, when $N_u \ll N_f$, and $\lambda$ is large such that $D_1^2 \ll D^2$, $D_0^2 \ll D^2$ and $\bar{I}_0 \gg \sigma^2$, the caching gain can be approximated as

$$G_{c1} \approx P_1^u \lambda B F, \tag{15}$$

which increases with $\lambda$ and $P_1^u$. Because the cache hit ratio $P_1^u$ first grows with $N_u$ rapidly but the increasing speed gradually becomes slower [11], $G_{c1}$ first grows with $N_u$ quickly and then saturates. This suggests that there is a tradeoff between the throughput gain from caching and the cache size at each user, which is consistent with the observation in [10], although file popularity is not considered and the transmission is assumed error-free in [10]. If all files in the catalog can be cached at each user, then the cache gain achieves an upper bound.

If $N_u \to N_f$, e.g., the catalog size $N_f$ in the SCN is not very large, most requests will be "served" by the users. Then, the real average arrival rate at BSs $\lambda_1 \to 0$, which results in $D_1 = D$. When $\lambda$ is large such that $D_0^2 \ll D^2$, we can obtain an approximate upper bound for the caching gain, which is

$$G_{c1}^U \approx \lambda B F - BW \frac{\alpha}{\ln 2}, \tag{16}$$

which only depends on and linearly increases with $\lambda$.

*2) Policy 2 (Prefetching):* Since prefetching can be analyzed as a special case of precaching, the caching gain of prefetching can be obtained as

$$G_{c2} = g(P_2^u, D_2), \tag{17}$$

where the function $g(\cdot)$ is defined in (14).

Analogically to precaching, when $N_u^s \ll N_u$, and $\lambda$ is large such that $D_2^2 \ll D^2$, $D_0^2 \ll D^2$ and $\bar{I}_0 \gg \sigma^2$, the caching gain can be approximated as

$$G_{c2} \approx P_2^u \lambda B F, \tag{18}$$

which increases with $\lambda$ and $P_2^u$. Again, since the cache hit ratio $P_2^u = r_u \sum_{j=1}^{N_u^s} j^{-\beta_s}/\sum_{i=1}^{N_s} i^{-\beta_s}$, $G_{c2}$ increases with $r_u$ linearly, and first grows with $N_u^s$ quickly and then saturates.

## IV. Numerical and Simulation Results

In this section, we verify the accuracy of the approximations in the derivation and evaluate the performance of different policies by numerical and simulation results.

In the simulation, we consider multiple small cells each with radius $D = 30$ m placed in an area with radius 250 m.

Then, $B = 61$. The path-loss model is $30.6 + 36.7\log_{10}(l)$ [14]. Each BS is with $M = 4$ antennas and transmit power $P = 23$ dBm. $W = 20$ MHz, $\sigma^2 = -95$ dBm. The file catalog $N_f = 10^4$ files, each of size $F = 30$ MBytes [3], $N_s = 100$ for prefetching and the parameters of Zipf distribution $\beta_s = \beta = 1$. Considering that in practice the storage capacity at today's smart phones can be 10-64 GBytes [3], we set the memory size for each user $C_u = 10$ GBytes and $N_u = 10$. Then, only $r_c = 0.3\%$ user's memory is used for precaching. The percentage of the users that can participate prefetching is set as $r_u = 50\%$, and $N_u^s = 1$. The simulation results are obtained by averaging over $10^5$ Rayleigh fading channel realizations, and the simulation time is 1000 s.

Although we assume that only one user can be served by each BS in previous analysis, multiuser scenario is considered in the simulation to validate the analysis, where each BS can serve at most $M$ users at the same time with equal power allocation and zero forcing beamforming. To ensure the system stable, we again employ admission control. When a BS is serving $M$ users and has users waiting in the queue, the newly arrived request will not be accessed, otherwise the request will be served or wait in the queue of the BS.

This setup is used in the sequel unless otherwise specified.

### A. Validation of Analysis and Performance Comparison

In Fig. 2(a), we compare the numerical results of the average throughput with the simulation results. The numerical results are directly computed from (6), (9) and (12), and the simulation results are provided for both single user and multi-user scenarios. We can see that the numerical and simulation results in single user scenario overlap, which indicates that the approximations used in deriving the average throughputs are accurate. The results in multi-user scenario are close to those in single user scenario, which means that previous analyses are also valid for multi-user scenario. When $\lambda$ is small, all policies achieve the same throughput, which linearly increases with $\lambda$. With the increase of traffic load, the throughput of baseline network first increases and then saturates due to severe ICI. By contrast, the throughput of precaching continues to grow with $\lambda$ thanks for the pre-offloading, and is about 200 % over the baseline network when $\lambda = 1.5$ files/s. The average throughput of prefetching is higher than the baseline but is much lower than precaching, due to the limited number of participated users and the number of cached files.

In Fig. 2(b), we compare the average file download time of different policies using simulation. The download time refers to the duration from a request arrived at a BS to the file completely downloaded to a user, which includes the queuing delay and file transmission time. If the requested file is already cached in a user, then the download time for this file is zero. When $\lambda$ is small, although all policies achieve the same throughput as in Fig. 2(a), the average download time can be reduced by precaching owing to the zero delay of the pre-downloaded files. With the increase of the traffic load, the reduced download time for precaching becomes more pronounced, where the download time is more than halved

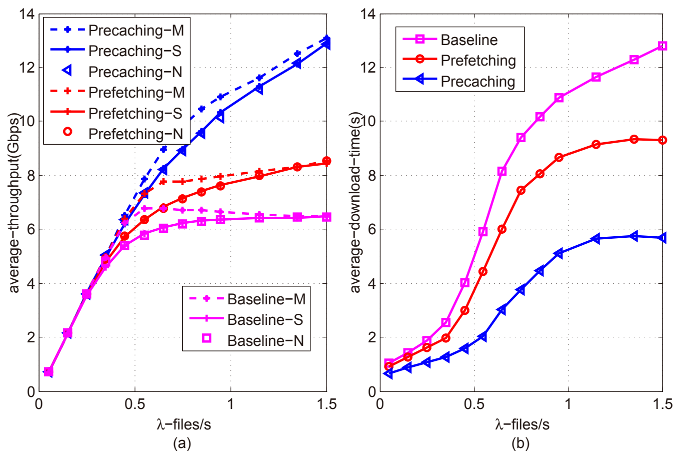

Fig. 2. Accuracy of the approximations and performance comparison. Legends: M-simulation for multi users scenario, S-simulation for single user scenario, N-numerical result of analysis

compared to the baseline when $\lambda = 1.5$ files/s. This indicates that even if some users may not request these cached files, in average the download time experienced by the users can be significantly reduced, which might justify the motivation that the users are willing to contribute a little fraction of their own memories to cache several popular files. Compared to precaching, the average download time of prefetching are much high, though also superior to the baseline.

### B. Caching Gain and Download Time Reduction

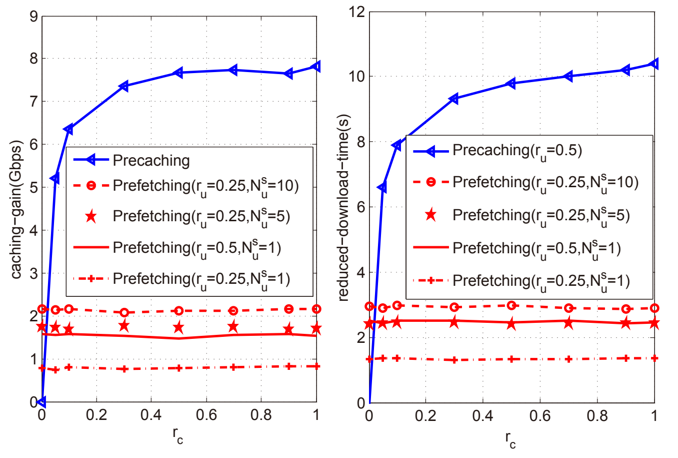

Fig. 3. Caching gain and reduced download time vs $r_c$, $\lambda = 1$ file/s, $r_u = 50\%$ and 25%, $N_u^s = 1$, 5 and 10.

In Fig. 3, we provide simulation results for the caching gain and download time reduction of precaching and prefetching versus the percentage of memory at each user occupied by precaching $r_c = N_u F/C_u$. The reduced download time is computed by subtracting the average download time of precaching (or precaching) from that of the baseline network. In the considered setting with $\lambda = 1$ file/s, most of BSs are busy to serve more than three users. It means that each BS can employ at most one antenna for pushing the files to user caches in prefetching. This implies that if excess resources of the same network is used for content placement, $r_u \leq 25\%$. When $r_u = 25\%$, we can see that the caching gain of prefetching is halved compared to $r_u = 50\%$ with given $N_u^s$, which agrees with the result in (18). Moreover, it is shown that the caching gain of precaching increases with $r_c$, but the growth speed becomes slower, which agrees with the analytical results in (15). This indicates that there is no need to employ much of local cache capacity of users to precaching popular files. By occupying a little cache capacity, precaching can offer remarkable benefit, both to the network and to the user. The simulation results show that if 25% of local memory of each user can be used by precaching, there will be about 750% caching gain over the baseline, 350% caching gain and 300% download time reduction with respect to prefeching with $r_u = 25\%$ and $N_u^s = 10$ (i.e., occupy the same cache capacity with precaching). Again, this provides a strong incentive for the users to participate precaching.

### C. Impact of File Popularity and Catalog Size

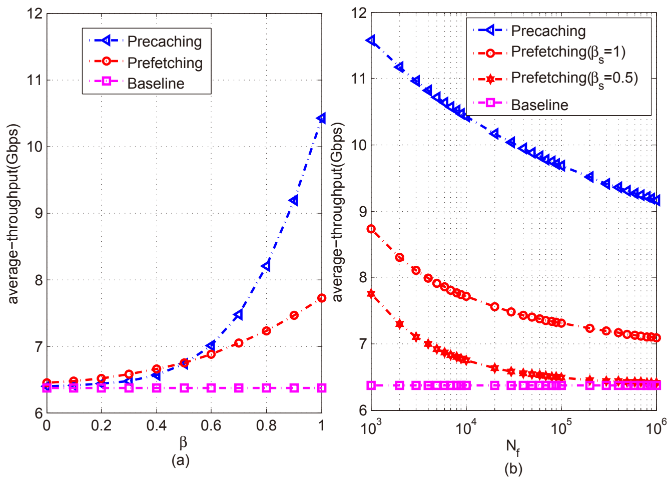

Fig. 4. Impact of file popularity and catalog size, $\lambda = 1$ files/s, in (b) $N_s = N_f/100$, $\beta = 1$.

In Fig. 4(a), we provide numerical results of the average throughput versus the Zipf distribution parameter, where $\beta = \beta_s$. It is shown that when the popularity distribution is uniform and the prediction in prefetching is not accurate, i.e., $\beta = \beta_s = 0$, the throughput gain from exploiting local cache at users is little, and all policies almost achieve the same performance. With the increase of $\beta$, both the caching gains of precaching and prefetching increase quickly, and the caching gain of precaching grows much faster.

In Fig. 4(b), we provide numerical results of the average throughput versus the catalog size $N_f$, where different prediction accuracy is considered for prefetching. As expected, the throughput decreases when $N_f$ increases. Nonetheless, even when the catalog size is $10^6$, where only $0.001/\%$ of all files in the catalog are cached at each user for the considered $N_u = 10$, the throughput can still be improved about 150% over the baseline. In fact, the catalog size is not the number of files available on the Internet. Instead, it is a function of the number of users in the network [3]. This means that the value of $N_f$ will not be too large in practice.

## V. Conclusions

In this paper, we quantified the performance gain of a simple policy of precaching, which proactively pre-downloads several popular files at the cache of each user based on the statistics of the user demands. We derived the average throughput for non-coordinated small cell networks with the precaching policy, and analyzed the caching gain over the network without caching. Analytical and simulation results showed that the throughput gain from caching is remarkable in high traffic load scenario attribute to alleviating interference by pre-offloading, and the gain largely depends on the popularity distribution. With an acceptable cache capacity, caching at each user can dramatically improve the average throughput of the network and reduce the average download time of the users by exploiting the content popularity. The precaching policy is superior to existing prefetching policy, especially when the popularity distribution is more "peaky". Such a gain is obtained without any coordination among small BSs and without the need to deploy any additional infrastructure.

## References

[1] N. Bhushan, J. Li, D. Malladi, R. Gilmore, D. Brenner, A. Damnjanovic, R. Sukhavasi, C. Patel, and S. Geirhofer, "Network densification: the dominant theme for wireless evolution into 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 82–89, 2014.
[2] H. Sun and S. A. Jafar, "Topological interference management with multiple antennas," in *IEEE ISIT*, 2014.
[3] 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 Commun. Mag.*, vol. 51, no. 4, pp. 142–149, 2013.
[4] M. Chen and A. Ksentini, "Cache in the air: exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, p. 132, 2014.
[5] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.
[6] B. D. Higgins, J. Flinn, T. J. Giuli, B. Noble, C. Peplin, and D. Watson, "Informed mobile prefetching," in *ACM MobiSys*, 2012.
[7] N. Gautam, H. Petander, and J. Noel, "A comparison of the cost and energy efficiency of prefetching and streaming of mobile video," in *ACM MoVid*, 2013.
[8] Y. Onoue, M. Tamai, and K. Yasumoto, "Energy-constrained Wi-Fi offloading method using prefetching," in *IEEE VTC Spring*, 2014.
[9] P. Lungaro, Z. Segall, and J. Zander, "Context-aware RRM for opportunistic content delivery in cellular networks," in *IEEE CTRQ*, 2010.
[10] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," in *IEEE ISIT*, 2013.
[11] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *IEEE INFOCOM*, 1999.
[12] D. P. Bertsekas, R. G. Gallager, and P. Humblet, *Data networks*. Prentice-Hall International New Jersey, 1992, vol. 2.
[13] R. W. Heath, T. Wu, Y. H. Kwon, and A. C. Soong, "Multiuser MIMO in distributed antenna systems with out-of-cell interference," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4885–4899, 2011.
[14] "Further advance-ments for E-UTRA physical layer aspects," 3GPP TR 36.814, Tech. Rep., 2010.