Fast Content Delivery via Distributed Caching and Small Cell Cooperation

Weng Chon Ao and Konstantinos Psounis, *Senior Member, IEEE*

Abstract—The demand for higher and higher wireless data rates is driven by the popularity of mobile video content delivery through wireless devices such as tablets and smartphones. To achieve unprecedented mobile content delivery speeds while reducing backhaul cost and delay, in this paper we propose a new system architecture that combines two recent ideas, distributed caching of content in small cells (FemtoCaching), and, cooperative transmissions from nearby base stations (Coordinated Multi-Point). A key characteristic of the proposed architecture is the interdependence between the caching strategy and the physical layer coordination. Specifically, the caching strategy may cache different content in nearby base stations (BSs) to maximize the cache hit ratio, or cache the same content in multiple nearby BSs such that the corresponding BSs can transmit concurrently, e.g. to multiple users using zero-forcing beamforming, and achieve multiplexing gains. Such interdependency allows a joint cross-layer optimization. Given the popularity distribution of the content, the available cache size, and the network topology, we devise optimal strategies of caching such that the system throughput is maximized or the system delay is minimized. Under realistic scenarios and assumptions, our analytical and simulation results show that our system yields significantly faster content delivery, which can be one order of magnitude faster than that of legacy systems.

1 Introduction

The popularity of mobile video streaming together with the proliferation of mobile devices such as smartphones and tablets are causing a tremendous growth of data traffic in cellular networks. To address this challenge the cellular industry is advocating a heterogeneous network architecture [1], [2] in which small cells (low power nodes), such as micro-BSs, pico-BSs, and femto-BSs, are deployed within traditional macrocells. These low power nodes provide short-range localized communication links resulting in a higher density of spatial reuse of radio recourses and thus in a higher overall network throughput.

There are many challenges in deploying a dense network of low power nodes. One such challenge that service providers consistently rank high is the deployment cost associated with connecting all the small cells to the backbone with fast links. Motivated by this, there is a growing interest to cache popular content to those low power nodes in a distributed manner, effectively trading off fast backhaul capacity with storage capacity. Specifically, the authors in [3]–[5] have introduced the concept of FemtoCaching, which is the idea of embedding femto-BSs with high storage capacity to store popular video files. When a user requests a video file, the user may be served by a nearby femto-BS that has the requested file in its cache over a high rate short-range wireless link. If the requested file is not in the cache of any nearby femto-BS, the user will be served directly by the macro-BS over a low rate long-range wireless link. Since the popularity distribution of video files changes at a much slower pace than that of user requests, cache updates (downloading popular video files via backhaul into the caches) can be done at off-peak hours, which results in a significant reduction of backhaul cost and delay while maintaining the performance benefits of a dense deployment of low power BSs.

Deploying a dense network of low power BSs yields even higher throughput when multiple neighboring BSs coordinate their data transmissions such that they aggregate constructively [6]–[10]. As a matter of fact, in the absence of such BS coordination, interference between nearby BSs may cancel the performance gains of dense deployments, and service providers consistently rank the technological challenges related to this issue as yet another major challenge in the deployment of small cells. There are many schemes for BS coordination, and in this paper we will consider the two most basic/popular ones: Maximum Ratio Transmission (MRT) and Zero-Forcing BeamForming (ZFBF) [11]. Consider that low power BSs form cooperation clusters. Then, under MRT, each BS in the cooperation cluster beamforms to a user such that the signals from the neighboring BSs are coherently combined, resulting in a diversity gain [12]. Under ZFBF, the BSs in the cluster simultaneously transmit multiple data streams to multiple users [10], [13], resulting in a multiplexing gain. Note that in the absence of offline cache updates, both MRT and ZFBF would further increase the cost and delay associated with backhaul, as they require multiple copies of the same files to be distributed to multiple BSs.

In this paper, we propose a new system architecture that combines FemtoCaching and femto-BS cooperation. The proposed cooperation scheme is cache-driven in the sense that if a typical user requests a video file, only the neighboring femto-BSs that have the requested video file in their caches will participate in the cooperative transmission. In other words, the cluster of cooperating femto-BSs is dynamically formed on a per-request basis. An important aspect of our system architecture is the joint cross-layer optimization of the cache allocation (content placement)

- *W. C. Ao and K. Psounis are with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA, 90089.*

in the application layer and the cooperative transmission techniques (MRT for diversity and ZFBF for multiplexing) in the physical layer. We jointly optimize these aspects of the system because caching different content in nearby caches increases hit ratio, but caching the same content increases the chances to get diversity and multiplexing gains. In general, the optimal cache allocation depends on a number of parameters, including the file popularity distribution, the cache size, the number of neighboring femto-BSs, and the transmission rate of the macro-BS in comparison to that of a femto-BS.

The remainder of this paper is organized as follows. We present related work in Section 2. Section 3 describes the setup, the caching strategies, and the cache-driven cooperation policies. In Section 4 we derive analytical formulas for the achieved rates under a variety of scenarios considering both background noise and co-channel interference. We extend the analysis to more complicated scenarios in Section 5. Section 6 analyzes the system performance from a non-saturation regime perspective by using queueing theory. In Section 7 we present numerical results for a number of real-world scenarios, highlighting the gains from our framework. Notably, our schemes can increase the speed of content delivery by an order of magnitude without requiring fast backhaul speeds. Last, Section 8 discusses practical considerations and Section 9 concludes the paper.

2 Prior work and contributions

This paper is related to a number of prior lines of work. Our setup is that of heterogeneous networks, formed by distributing multi-tier low power nodes (e.g., micro-BSs and femto-BSs) in macro-cellular networks, see, for example, two recent tutorial-style papers and references therein [1], [2]. It is building upon prior work on BS cooperation, more generally known as Coordinated Multi-Point (CoMP), and FemtoCaching. There is a long line of research in CoMP, see, for example, [6], [7]. FemtoCaching has been recently introduced to trade off backhaul capacity with cache capacity [3]–[5] and can be further applied to device-to-device communication networks [14], [15] and to coded caching [16], [17]. FemtoCaching itself is building upon prior work on distributed caching, content placement schemes and content distribution networks, see, for example, [18].

In addition to using standard analytical tools like convex and integer optimization, combinatorics, and Shannon rate formulas, we also use stochastic geometry [19] to take into account co-channel interference in the context of heterogeneous networks, see, for example, the relevant analysis in [20].

Directly related to this work is [21], [22] where the authors use a coding scheme to introduce redundancy in caches and create CoMP opportunities for cooperative transmissions. A fundamental difference between this prior work and our paper is that we consider the effect of cache misses, since any type of redundancy decreases the number of distinct files that can be stored in finite size caches. To optimize the system performance, we appropriately control the stored redundancy for each individual file and dynamically (per-request basis) form a cluster of cooperating femto-BSs.

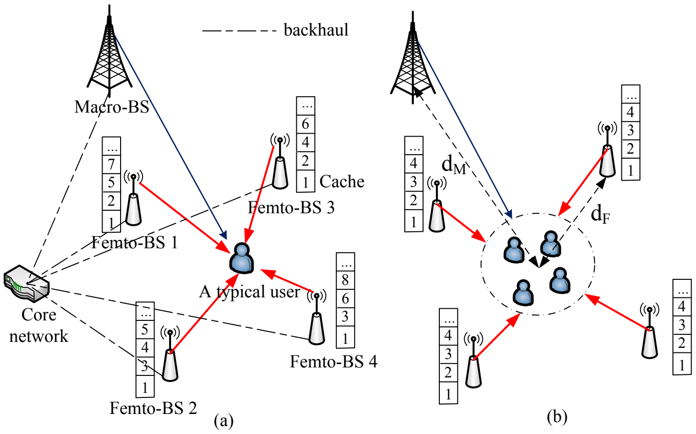


Fig. 1: System model for cache-driven femto-BS cooperation: (a) randomized caching and MRT (b) threshold-based caching and ZFBF.

Our contributions are as follows: We combine the concepts of FemtoCaching and BS cooperation to propose a novel, high-performing system architecture. We derive analytical expressions for the user rates and delays and jointly optimize the caching strategy and the PHY layer cooperation. We devise efficient caching strategies for providing diversity gains under MRT, multiplexing gains under ZFBF, and the optimal diversity-multiplexing tradeoff. Last, we study the performance of our schemes under practical scenarios and address deployment considerations.

3 System model

3.1 Topology

Consider a typical user in a macrocell. Suppose there are N femto-BSs and another $K-1$ users in the neighborhood of the typical user. We denote by $d_{0,k}$, $1 \leq k \leq K$, and $d_{j,k}$, $1 \leq j \leq N$, $1 \leq k \leq K$, the distance between the macro-BS and the kth user, and the jth femto-BS and the kth user respectively. Let these K co-located users be associated with the same N neighboring femto-BSs and the macro-BS. These N neighboring femto-BSs are candidates for cooperative transmissions, see Fig. 1a for a scenario where femto-BSs transmit the same content (say, file 1) to one user, and Fig. 1b for a scenario where femto-BSs transmit concurrently to multiple users (say, to four users four different files, namely file 1, 2, 3, and 4).

In a typical real-world scenario one may have tens or hundreds of femto-BSs inside a macrocell and hundreds or thousands of users. Thus, femto-BSs would be grouped into clusters of nearby femto-BSs which can concurrently serve a number of users. For example, one may have one such cluster per floor on a large building or one cluster per building. In the rest of this paper we will focus on one such cluster of co-located femto-BSs and users.[1]

In general, the users, the macro-BS, and the femto-BSs would be assigned different time-frequency slots (resource blocks) for data transmissions. We will only focus on the downlink, and thus on transmissions towards the K users.

[1] It is beyond the scope of this paper to further investigate clustering algorithms.

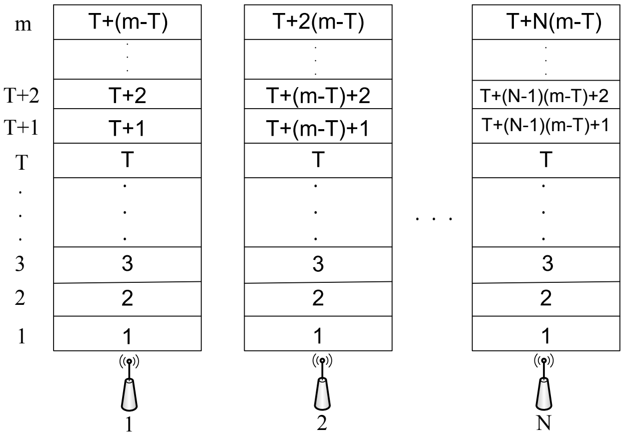


Fig. 2: Caching strategy under threshold-based caching and ZFBF.

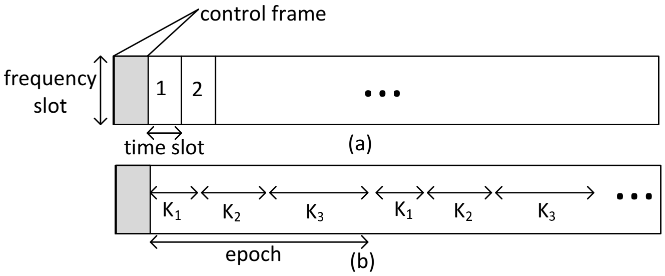


Fig. 3: Control and data frames under (a) randomized caching and MRT, (b) threshold-based caching and ZFBF.

Also, for simplicity and without loss of generality, we will focus on a single frequency slot which the macro-BS and the femto-BSs may use to transmit to the users. In some of the physical layer schemes that we study, one time slot can be used for a single user only, e.g. MRT, while in others it can be used for multiple users concurrently, e.g. ZFBF. Depending on whether a user is served by the macro-BS, a single femto-BS, or multiple femto-BSs using MRT or ZFBF, the data rate that it will receive during this time slot will vary. Last, a control frame will be used to collect requests from the users at the macro-BS, and then the users will be served in subsequent time slots.

3.2 Caching strategies and cache-driven cooperation policies

Suppose that there is a library of M video files which are ordered according to their (normalized) popularity $p_i,\ 1 \leq i \leq M$, $p_1 \geq p_2 \geq \cdots \geq p_M$, $\sum_{i=1}^{M} p_i = 1$. In other words, a typical user would request the ith file with probability p_i. Furthermore, we define the cdf of the file popularity distribution as $v_j \triangleq \sum_{i=1}^{j} p_i,\ j = 1, \cdots, M$. To simplify the analytical exposition and without loss of generality, we assume that all files have the same size and the unit of cache size is a file.[2] The macro-BS stores (or has access to) all of the M files, while each femto-BS has a cache that can store on average up to m files with $m < M$ (i.e., the average cache size of a femto-BS is m). We have the following two different caching and cooperation strategies aiming at providing diversity gains and multiplexing gains.

3.2.1 *Providing diversity gains by randomized caching and MRT*

For each femto-BS the caching strategy caches the ith file with probability $q_i,\ 0 \leq q_i \leq 1,\ i = 1, \cdots, M$ subject to the probabilistic cache size constraint $\sum_{i=1}^{M} q_i \leq m$. Note that $q_i,\ i = 1, \cdots, M$ are design parameters.

Then, the cooperation policy dictates that for a request of the ith file, generated by a typical user according to the popularity distribution p_i, the request will be jointly served by the femto-BSs that have the requested file in their caches. Thus, no data file exchanges are needed for the cooperation.[3] Under MRT, each femto-BS that has the requested file in its cache beamforms its signal to the typical user so that the signals (from the cooperating femto-BSs) are coherently combined at the receiver, producing a diversity gain for the desired signal. If none of the femto-BSs have the requested file, the request will be served by the macro-BS. For example, in Fig. 1a, if a user requests the 2nd file, then femto-BSs 1 and 3 will beamform to the user.

Fig. 3a shows a typical sequence of time slots for the randomized caching and MRT scheme, where each file request is served for one time slot.

3.2.2 *Providing multiplexing gains by threshold-based caching and ZFBF*

Aiming at providing multiplexing gains we consider the following caching strategy. We first choose a threshold T, $0 \leq T \leq m$, which is a design parameter (we will generalize to the case with multiple thresholds in Section 5.1). We cache the files 1 to T (referred to as type 1 files) in all of the femto-BSs, and cache the files $T+1$ to $T+N(m-T)$ (referred to as type 2 files) in exactly one of the femto-BSs (see Fig. 2). That is, we have N copies for each of the most popular files 1 to T, one copy of the files $T+1$ to $T+N(m-T)$, and the remaining files $T+N(m-T)+1$ to M (referred to as type 3 files) can be downloaded only by the macro-BS. As a result, the probability of a typical file request being type 1 is v_T. The probability of a typical file request being type 2 is $v_{T+N(m-T)} - v_T$. The probability of a typical file request being type 3 is $1 - v_{T+N(m-T)}$.

Since all the femto-BSs cache type 1 files, the N femto-BSs can coordinate their transmissions when serving type 1 file requests. Specifically, the N femto-BSs can potentially serve N type 1 file requests simultaneously, producing a multiplexing gain of order N.[4] No cooperative transmission is used for serving a type 2 (type 3) file request because only a single femto-BS (the macro-BS) caches the requested file.

Fig. 3b shows a typical sequence of epochs for the threshold-based caching and ZFBF scheme, where an epoch is defined as a collection of time slots of length $K_1+K_2+K_3$. In the following analysis we will normalize the epoch length to one, i.e., $K_1 + K_2 + K_3 = 1$. Thus, $K_i,\ i = 1, 2, 3$, will

[2] This is different from the basic unit for data transmission (bits).

[3] Control signals such as channel state information may be required to be distributed among the femto-BSs in the cooperation cluster.

[4] By using ZFBF, suppose there are N transmitters and K receivers ($K \leq N$), the N transmitters can transmit K independent (non-interfering or spatially isolated) streams to the K receivers simultaneously, each with a diversity of order 1.

TABLE 1: Main notations

Number of femto-BSs	N
Number of antennas in a (macro)femto-BS	L_M (L_F)
Distance between a typical user and a macro(femto)BS	d_M (d_F)
Number of video files	M
Cache size in a femto-BS	m
Number of users	K
File popularity distribution	p_i (cdf v_i)
File size	L
Probability of caching the ith file for MRT	q_i
Caching threshold for ZFBF	T
Time portion for serving type i files	K_i
Bandwidth of a frequency slot	W
Transmit power of macro(femto)-BS	P_M (P_F)
Data rate of macro(femto)-BS	R_M (R_F)
Effective data rate of macro(femto)-BS	$\tilde{R}_M$ ($\tilde{R}_F/\tilde{R}_F^{(j)}$)
Path loss exponent	α
Noise power spectral density	N_0
Density of interfering macro(femto)-BSs	λ_M (λ_F)
Average effective data rate with MRT(ZFBF)	$R_{\text{MRT}}(R_{\text{ZFBF}})$
Average effective data rate with MRT–ZFBF	$R_{\text{MRT–ZFBF}}$
Average service time with MRT(ZFBF)	$D_{\text{MRT}}(D_{\text{ZFBF}})$
Arrival rate of file requests	λ
Average system delay with MRT(ZFBF)	$W_{\text{MRT}}(W_{\text{ZFBF}})$

be the portion of time allocated for transmitting type i files. (See Section 4.2.)

4 Performance analysis

We study the saturation throughput, that is, we assume that there are always enough requests for files, and thus enough pending bits, to be transmitted to the various users at every time slot.

For simplicity, we assume that the distances between the co-located users and the N neighboring femto-BSs (the macro-BS) are the same, i.e., $d_{j,k} \triangleq d_F$ ($d_{0,k} \triangleq d_M$), see Fig. 1b for a pictorial representation. Also, we assume that the macro-BS, the femto-BSs, and the users are all equipped with a single antenna (see Section 5.2 for the case with multiple antennas). We consider quasi-static Rayleigh flat-fading channels with unit mean power. We denote the transmit power of a macro-BS and a femto-BS by P_M and P_F, and the data rates of a macro-BS and a femto-BS by R_M and R_F. The bandwidth of the frequency slot is denoted by W, the path loss exponent by α, and the noise power spectral density by N_0.

To simplify the notation in the following derivations, we also define the effective data rate as the data rate multiplied by the non-outage (transmission success) probability (i.e., the probability that the channel can support the data rate [11]). Thus, for a macro-BS the effective data rate equals $\tilde{R}_M \triangleq R_M \cdot \Pr\left(W\log\left(1+\frac{P_M S_{\Gamma(1)} d_M^{-\alpha}}{N_0 W}\right) > R_M\right)$ and for a femto-BS cluster with a diversity of order j it equals $\tilde{R}_F^{(j)} \triangleq R_F \cdot \Pr\left(W\log\left(1+\frac{P_F S_{\Gamma(j)} d_F^{-\alpha}}{N_0 W}\right) > R_F\right)$, where $S_{\Gamma(1)}$ is an exponential random variable with unit mean (Rayleigh fading), and $S_{\Gamma(j)}$ is the sum of j i.i.d. exponential random variables with unit mean due to the coherent combining of the signals from j femto-BSs. Note that for further simplicity, when $j=1$ we denote $\tilde{R}_F^{(1)}$ as $\tilde{R}_F$. Table 1 summarizes the main notations used in the paper.

In the following, convex optimization is used in the performance analysis of randomized caching and MRT, while integer optimization is used in the performance analysis of threshold-based caching and ZFBF. Also, in Sections 4.1-4.3, we consider background noise without co-channel interference, and Section 4.4 extends the results in the presence of co-channel interference.

4.1 Randomized caching and MRT

We aim to maximize the average effective data rate of a typical user (denoted as R_{MRT}) with respect to the randomized caching parameters subject to the cache size constraint.

We have the following optimization problem:

$$\begin{aligned} \underset{q_1,\cdots,q_M}{\text{maximize}} \quad & \sum_{i=1}^{M} p_i U(q_i) \triangleq R_{\text{MRT}} \\ \text{subject to} \quad & \sum_{i=1}^{M} q_i \le m, \\ & 0 \le q_i \le 1,\ i=1,\cdots,M, \end{aligned} \tag{1}$$

where p_i is the file popularity distribution (i.e., the probability that the typical user requests the ith file), q_i is the probability that the ith file is cached in a femto-BS, and $U(q_i)$ is the effective data rate for the transmission of the ith file. (See Section 3.2.1 for a detailed description of the caching strategy.) Since we have N femto-BSs, the number of copies of the ith file in the femto-BSs is a binomial random variable with mean Nq_i. Thus, the number of cooperating femto-BSs for the transmission of the ith file (denoted as C_i) follows the law

$$\Pr(C_i = j) = \binom{N}{j} q_i^j (1-q_i)^{N-j},\ j=0,1,\cdots,N. \tag{2}$$

The effective data rate for the transmission of the ith file, $U(q_i)$, can be computed as

$$\begin{aligned} U(q_i) &= \Pr(C_i=0) R_M \Pr(\text{transmission success with macro-BS}) \\ &+ \sum_{j=1}^{N} \Pr(C_i=j) R_F \Pr(\text{trans. success with } j \text{ femto-BSs jointly}) \\ &= (1-q_i)^N \tilde{R}_M + \sum_{j=1}^{N} \binom{N}{j} q_i^j (1-q_i)^{N-j} \tilde{R}_F^{(j)}. \end{aligned} \tag{3}$$

Since we have assumed Rayleigh fading, we can rewrite the effective data rates as $\tilde{R}_M = R_M e^{-\eta_M}$ and $\tilde{R}_F = R_F e^{-\eta_F}$ where $\eta_M = (2^{R_M/W}-1)N_0 W d_M^{\alpha}/P_M$ and $\eta_F = (2^{R_F/W}-1)N_0 W d_F^{\alpha}/P_F$. We have the following theorem.

Theorem 1. *If $R_M e^{-\eta_M} \le R_F e^{-\eta_F}(1-\eta_F)$, then $U(q_i)$ is concave and thus $R_{MRT}(q_1,\cdots,q_M)$ is concave in the domain $0 \le q_i \le 1,\ i=1,\cdots,M$.*

Proof. The proof is provided in Appendix. □

The condition $R_M e^{-\eta_M} \leq R_F e^{-\eta_F}(1-\eta_F)$ implies that the marginal rate gain of including one more femto-BS into the cluster to perform cooperative transmission is decreasing and is smaller than the difference between the rates of a femto-BS and a macro-BS (see Appendix for a detailed discussion). These conditions usually hold in practice[5].

For the maximization problem with concave objective and linear constraints, we can find the optimal q_i^* by the KKT conditions [23]. We have:

$$q_i^* = \left[U'^{-1}\left(\frac{\mu}{p_i}\right)\right]_0^1, \; i = 1, \cdots, M;$$
$$\sum_{i=1}^{M} q_i^* = m, \tag{4}$$

where μ is the Lagrange multiplier, $U'^{-1}(\cdot)$ is the inverse function of $U'(\cdot)$, which exists since $U'(\cdot)$ is monotone (see Lemma 1 in Appendix), and $[x]_0^1 \triangleq \min(\max(0,x),1)$. We also note that $U'^{-1}(\cdot)$ is a decreasing function. So, q_i^* increases as p_i increases, i.e., we have a higher probability to cache a more popular file, which makes sense intuitively.

4.2 Threshold-based caching and ZFBF

Recall that we are considering the saturation throughput with unlimited pending file requests for each user. We first consider the case where the file requests are generated by K users with $K \geq N$, thus the maximum multiplexing gain of N is achievable. Latter in the subsection we also consider the case where $K < N$, which restricts the multiplexing gain to K. Also, recall that type 1 files are cached in all N femto-BSs, while type 2 (type 3) files are only cached in a single femto-BS (the macro-BS). (See Section 3.2.2 for a detailed description of the caching strategy.) By using ZFBF, we can simultaneously transmit N type 1 files with sum rate $N\tilde{R}_F$, while the transmission rate for a type 2 (type 3) file is $\tilde{R}_F$ ($\tilde{R}_M$).

We divide an epoch into three portions with durations K_1, K_2, and K_3 for serving the type 1 file requests, type 2 file requests, and type 3 file requests respectively (see Fig. 3b). Without loss of generality, the epoch length is normalized to 1, i.e., $K_1+K_2+K_3 = 1$. We choose to allocate system resources to the three types of files as follows: Type 1 files represent a portion of files equal to v_T and are served with rate $N\tilde{R}_F$ for K_1 portion of time, yielding an effective traffic load of $\frac{v_T}{K_1 N\tilde{R}_F}$. Similarly we compute the effective traffic load of type 2 and type 3 files, and we equate the three traffic loads, essentially treating all three types equally. Thus, the values of K_1, K_2, and K_3 satisfy the conservation law:

$$K_1 N\tilde{R}_F : K_2\tilde{R}_F : K_3\tilde{R}_M$$
$$= v_T : v_{T+N(m-T)} - v_T : 1 - v_{T+N(m-T)}. \tag{5}$$

Therefore, we have

$$K_1 = \frac{\frac{v_T}{N\tilde{R}_F}}{\frac{v_T}{N\tilde{R}_F} + \frac{v_{T+N(m-T)}-v_T}{\tilde{R}_F} + \frac{1-v_{T+N(m-T)}}{\tilde{R}_M}},$$
$$K_2 = \frac{\frac{v_{T+N(m-T)}-v_T}{\tilde{R}_F}}{\frac{v_T}{N\tilde{R}_F} + \frac{v_{T+N(m-T)}-v_T}{\tilde{R}_F} + \frac{1-v_{T+N(m-T)}}{\tilde{R}_M}},$$
$$K_3 = \frac{\frac{1-v_{T+N(m-T)}}{\tilde{R}_M}}{\frac{v_T}{N\tilde{R}_F} + \frac{v_{T+N(m-T)}-v_T}{\tilde{R}_F} + \frac{1-v_{T+N(m-T)}}{\tilde{R}_M}}. \tag{6}$$

The average effective data rate, denoted as $R_{\text{ZFBF}}(T)$ where T is the caching threshold, can be computed as

$$R_{\text{ZFBF}}(T) = K_1 N\tilde{R}_F + K_2\tilde{R}_F + K_3\tilde{R}_M \tag{7}$$
$$= \frac{1}{\frac{v_T}{N\tilde{R}_F} + \frac{v_{T+N(m-T)}-v_T}{\tilde{R}_F} + \frac{1-v_{T+N(m-T)}}{\tilde{R}_M}}.$$

Next, we wish to maximize the average effective data rate by choosing a proper caching threshold T. We can find the optimal caching threshold T^* by enumeration of the solution space $T = 0, 1, \cdots, m$, which is linear in the cache size m and thus easy to compute in practice. Furthermore, we can analytically solve for the optimal T^* when the discrete file popularity distribution can be approximated by a continuous one. For example, consider the Zipf distribution with parameter s, i.e., $p_i = c_{M,s}/i^s, \; i = 1, \ldots, M$, where $c_{M,s}$ is the normalization constant. We approximate v_T and $v_{T+N(m-T)}$ by

$$v_T = \sum_{i=1}^{T} p_i = c_{M,s}\sum_{i=1}^{T}\frac{1}{i^s} \approx c_{M,s}\int_1^T \frac{1}{x^s}dx,$$
$$v_{T+N(m-T)} \approx c_{M,s}\int_1^{T+N(m-T)} \frac{1}{x^s}dx. \tag{8}$$

Substituting Eq. (8) into Eq. (7) and then differentiating $R_{\text{ZFBF}}(T)$ with respect to T, the optimal threshold can be obtained as

$$T^* \approx \left[\frac{\xi N m}{1+\xi(N-1)}\right]_0^m, \text{ where } \xi \triangleq \left(\frac{\tilde{R}_M}{N(\tilde{R}_F - \tilde{R}_M)}\right)^{\frac{1}{s}}. \tag{9}$$

(The notation $[x]_0^m \triangleq \min(\max(0,x),m)$.) Note that the optimal threshold T^* is proportional to the cache size m. Fig. 8 in Section 7 shows the resulting data rate under various threshold values for practical scenarios. **Example:** We consider two special cases.

- When $T = m$, we cache the most popular m files in all N femto-BSs and we have $R_{\text{ZFBF}}(m) = \frac{1}{\frac{v_m}{N\tilde{R}_F}+\frac{1-v_m}{\tilde{R}_M}}$. If $v_m \approx 1$, that is, the m most popular files contain most probability mass, we have $R_{\text{ZFBF}}(m) \approx N\tilde{R}_F$.
- When $T = 0$, we cache only one copy of the most popular Nm files in femto-BSs and we have $R_{\text{ZFBF}}(0) = \frac{1}{\frac{v_{Nm}}{\tilde{R}_F}+\frac{1-v_{Nm}}{\tilde{R}_M}}$. If $v_{Nm} \approx 1$, we have $R_{\text{ZFBF}}(0) \approx \tilde{R}_F$.
- We can see that for very skewed popularity distribution satisfying $v_m \approx 1$, the rate R_{ZFBF} with threshold $T = m$ is N times higher than that with threshold $T = 0$, where N is the maximum multiplexing gain.

We now study the case where the number of users K can be smaller than the number of femto-BSs N. When $K <$

[5] When these assumptions do not hold, one can still solve the optimization problem using non-convex optimization techniques.

N, the maximum multiplexing gain is K, limited by the number of users. Thus, the effective traffic load of type 1 files is now equal to $\frac{v_T}{K_1 \min\{K,N\}\tilde{R}_F}$ and K_1, K_2, K_3 must satisfy

$$\begin{aligned} &K_1 \min\{K,N\}\tilde{R}_F : K_2\tilde{R}_F : K_3\tilde{R}_M \\ &\quad = v_T : v_{T+N(m-T)} - v_T : 1 - v_{T+N(m-T)}. \end{aligned} \tag{10}$$

Similarly to before, after solving for K_1, K_2, and K_3, we have

$$\begin{aligned} R_{\text{ZFBF}}(T) &= K_1 \min\{K,N\}\tilde{R}_F + K_2\tilde{R}_F + K_3\tilde{R}_M \qquad (11) \\ &= \frac{1}{\frac{v_T}{\min\{K,N\}\tilde{R}_F} + \frac{v_{T+N(m-T)}-v_T}{\tilde{R}_F} + \frac{1-v_{T+N(m-T)}}{\tilde{R}_M}}. \end{aligned}$$

Also, under the continuous approximation of the Zipf distribution, we compute the optimal caching threshold to be

$$T^* \approx \left[\frac{\xi N m}{1+\xi(N-1)}\right]_0^m, \; \xi \triangleq \left(\frac{\frac{1}{\min\{K,N\}\tilde{R}_F} - \frac{1}{\tilde{R}_F}}{(N-1)(\frac{1}{\tilde{R}_F} - \frac{1}{\tilde{R}_M})}\right)^{\frac{1}{s}}. \tag{12}$$

Remark: It is evident that there is a tradeoff associated with the value of the design parameter T. When T is large, we benefit from the multiplexing gain but more redundant files are held in the caches, resulting in an increasing amount of requests towards the low-rate macro-BS (cache misses). When T is small, we lose the multiplexing gain but most of the files are in the caches of the femto-BSs, generating fewer requests towards the macro-BS. The optimal choice of T depends on the file popularity distribution.

4.3 Joint MRT–ZFBF

It is known that both diversity and multiplexing gains can be achieved by a careful design of the ZFBF precoding [24]. Specifically, suppose there are N transmitters and K receivers ($K \leq N$). Then, the N transmitters can transmit K independent (non-interfering or spatially isolated) streams to the K receivers simultaneously, each with a diversity of order $N-K+1$ [24]. We refer to this scheme as the MRT–ZFBF scheme. Assuming the threshold-based caching and following the analysis in Section 4.2 we define K_1, K_2, and K_3 such that $K_1+K_2+K_3=1$ and

$$\begin{aligned} &K_1 \min\{K,N\}\tilde{R}_F^{(N-\min\{K,N\}+1)} : K_2\tilde{R}_F : K_3\tilde{R}_M \\ &\quad = v_T : v_{T+N(m-T)} - v_T : 1 - v_{T+N(m-T)}. \end{aligned} \tag{13}$$

The average effective date rate of the MRT–ZFBF scheme (denoted as $R_{\text{MRT–ZFBF}}(T)$) can be computed as

$$\begin{aligned} &R_{\text{MRT–ZFBF}}(T) = \\ &K_1 \min\{K,N\}\tilde{R}_F^{(N-\min\{K,N\}+1)} + K_2\tilde{R}_F + K_3\tilde{R}_M, \qquad (14) \\ &= \frac{1}{\frac{v_T}{\min\{K,N\}\tilde{R}_F^{(N-\min\{K,N\}+1)}} + \frac{v_{T+N(m-T)}-v_T}{\tilde{R}_F} + \frac{1-v_{T+N(m-T)}}{\tilde{R}_M}}. \end{aligned}$$

When $K < N$, each user gets a diversity of order $N-K+1$, receiving a type 1 file at a rate $\tilde{R}_F^{(N-K+1)}$. Under the continuous approximation of the Zipf distribution, the optimal caching threshold equals

$$T^* \approx \left[\frac{\xi N m}{1+\xi(N-1)}\right]_0^m, \; \xi \triangleq \left(\frac{\frac{1}{\min\{K,N\}\tilde{R}_F^{(N-K+1)}} - \frac{1}{\tilde{R}_F}}{(N-1)(\frac{1}{\tilde{R}_F} - \frac{1}{\tilde{R}_M})}\right)^{\frac{1}{s}}. \tag{15}$$

4.4 Noise and co-channel interference

We extend the analytical model to take into account co-channel interference from other macro-BSs and other femto-BSs outside the cooperation cluster using the same frequency slot at the same time with the typical user under study (i.e., the same resource block is spatially reused). We can approximate the spatial distribution of the interfering macro-BSs (outside the circle centered at the typical user with radius d_M, denoted as $B(0,d_M)$) as a Poisson point process Φ with some density λ_M [20]. The co-channel interference from the interfering macro-BSs can be written as I_M, $I_M \triangleq \sum_{x\in\Phi\setminus B(0,d_M)} P_M S_{x,\Gamma(1)} D_x^{-\alpha}$, where $S_{x,\Gamma(1)}$ denotes the channel gain for the small-scale fading of the interference from the xth interfering macro-BS to the typical user, which is exponentially distributed with unit mean (Rayleigh fading). D_x is the distance between the xth interfering macro-BS and the typical user.

Similarly, we can approximate the spatial distribution of the interfering femto-BSs (outside the circle centered at the typical user with radius d_F, denoted as $B(0,d_F)$) as a Poisson point process Ψ with some density λ_F. The co-channel interference from the interfering femto-BSs can be written as I_F, $I_F \triangleq \sum_{y\in\Psi\setminus B(0,d_F)} P_F S_{y,\Gamma(1)} D_y^{-\alpha}$, where $S_{y,\Gamma(1)}$ and D_y are similarly defined.

Following the derivations in [25], the success (non-outage) probability of the transmission between the typical user and its serving macro-BS can be computed as

$$\begin{aligned} &\Pr\left(W\log\left(1+\frac{P_M S_{\Gamma(1)} d_M^{-\alpha}}{I_M+I_F+N_0W}\right) > R_M\right) \\ &= \exp\left(-\tau_M N_0 W d_M^{\alpha} P_M^{-1}\right)\exp\left\{-\lambda_M \pi d_M^2\right. \\ &\left.\cdot\left(\tau_M^{\delta}\mathbb{E}_S[S^{\delta}\gamma(1-\delta,\tau_M S)] - \mathbb{E}_S[1-\exp(-\tau_M S)]\right)\right\} \\ &\cdot\exp\left\{-\lambda_F\pi d_F^2\left(\tau_M^{\delta}\eta^{\delta}\mathbb{E}_S[S^{\delta}\gamma(1-\delta,\tau_M\eta S)]\right.\right. \\ &\left.\left.-\mathbb{E}_S[1-\exp(-\tau_M\eta S)]\right)\right\}, \end{aligned} \tag{16}$$

where $\tau_M \triangleq 2^{R_M/W}-1$, $\delta \triangleq 2/\alpha$, $\eta \triangleq (\frac{d_M}{d_F})^{\alpha}\frac{P_F}{P_M}$, S is an exponential random variable with unit mean, and $\gamma(a,z) \triangleq \int_0^z \exp(-t)t^{a-1}dt$ is the lower incomplete gamma function.

Similarly, the success probability of the transmission between the typical user and a cluster of j femto-BSs can be computed as

$$\begin{aligned} &\Pr\left(W\log\left(1+\frac{P_F S_{\Gamma(j)} d_F^{-\alpha}}{I_M+I_F+N_0W}\right) > R_F\right) \\ &= \sum_{k=0}^{j-1}\frac{1}{k!}(-1)^k\frac{d^k}{dt^k}V(t)\bigg|_{t=1}, \end{aligned} \tag{17}$$

where $V(t)$ is defined as

$$V(t) \triangleq \exp\left(-\tau_F N_0 W d_F^{\alpha} t P_F^{-1}\right) \exp\Big\{-\lambda_F \pi d_F^2 \cdot \Big(\tau_F^{\delta} t^{\delta} \mathbb{E}_S[S^{\delta}\gamma(1-\delta,\tau_F tS)] - \mathbb{E}_S[1-\exp(-\tau_F tS)]\Big)\Big\} \cdot \exp\Big\{-\lambda_M \pi d_M^2 \Big(\tau_F^{\delta} t^{\delta} \eta^{-\delta} \mathbb{E}_S[S^{\delta}\gamma(1-\delta,\tau_F t\eta^{-1}S)] -\mathbb{E}_S[1-\exp(-\tau_F t\eta^{-1}S)]\Big)\Big\} \tag{18}$$

and $\tau_F \triangleq 2^{R_F/W} - 1$. With these expressions, we can compute the effective data rates following the same steps as we did in Section 4.1–4.3.

5 Extensions

5.1 Multiple thresholds

The threshold-based caching scheme in Section 4.2 can be generalized to one with multiple thresholds where the basic idea is that the more popular a file is, the larger the number of copies of the file is in the caches. Without loss of generality, assume that the number of neighboring femto-BSs is $N = 2^n$ and define n thresholds $T_0, T_1, \cdots, T_{n-1}$, which are design parameters. The T_0 most popular files will be stored in all 2^n femto-BSs like before (2^n copies each), the next $2T_1$ most popular files will have 2^{n-1} copies each, and in general, the threshold T_i means that we allocate $2^n T_i$ storage units to cache $2^i T_i$ files, each with 2^{n-i} copies. Fig. 4a shows an example with $n = 2$ ($N = 4$).

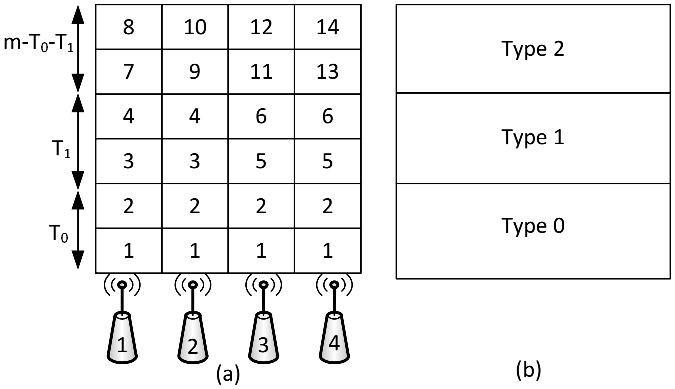


Fig. 4: Threshold-based caching and ZFBF with multiple thresholds ($m = 6,\ T_0 = 2,\ T_1 = 2,\ N = 2^n = 4$).

We partition the files into $n+2$ types (namely, type 0 files, type 1 files, $\cdots$, type $n+1$ files). Type i, $i = 0, 1, \cdots, n-1$ files refer to the $2^i T_i$ files stored in the caches with 2^{n-i} copies each (see Fig. 4b). Type n files refer to the $2^n \left(m - \sum_{i=0}^{n-1} T_i\right)$ files stored in the caches with a single copy each. Type $n+1$ files refer to the files only stored in the macro-BS. As a result, the probability of a typical file request being type i, denoted by a_i, is

$$\begin{aligned} a_0 &\triangleq v_{T_0}; \\ a_i &\triangleq v_{\sum_{j=0}^{i} 2^j T_j} - v_{\sum_{j=0}^{i-1} 2^j T_j},\ i = 1, \ldots, n-1; \\ a_n &\triangleq v_{\sum_{j=0}^{n-1} 2^j T_j + 2^n (m - \sum_{j=0}^{n-1} T_j)} - v_{\sum_{j=0}^{n-1} 2^j T_j}; \\ a_{n+1} &\triangleq 1 - a_0 - a_1 - \cdots - a_n. \end{aligned} \tag{19}$$

By using ZFBF, we can simultaneously transmit $\min\{K, 2^{n-i}\}$ type i files with sum rate $\min\{K, 2^{n-i}\}\tilde{R}_F,\quad i = 0, 1, \ldots, n$. The transmission rate for a type $n+1$ file is $\tilde{R}_M$.

We divide an epoch into $n+2$ portions with durations $K_i,\ i = 0, 1, \ldots, n+1$ for serving the type i file requests. Following the same rational as before, the epoch length is normalized to 1, i.e., $\sum_{i=0}^{n+1} K_i = 1$, and the values of $K_i,\ i = 0, 1, \ldots, n+1$ satisfy the conservation law:

$$K_0 \min\{K, 2^n\}\tilde{R}_F : K_1 \min\{K, 2^{n-1}\}\tilde{R}_F : \cdots : K_{n+1}\tilde{R}_M = a_0 : a_1 : \cdots : a_{n+1}. \tag{20}$$

The average effective data rate for the threshold-based caching and ZFBF scheme with multiple thresholds (denoted as $R_{\text{ZFBF}}^{\text{multi}}$) can be computed as

$$R_{\text{ZFBF}}^{\text{multi}}(T_0, \cdots, T_{n-1}) = K_{n+1}\tilde{R}_M + \sum_{i=0}^{n} K_i \min\{K, 2^{n-i}\}\tilde{R}_F = \frac{1}{\frac{a_{n+1}}{\tilde{R}_M} + \sum_{i=0}^{n} \frac{a_i}{\min\{K, 2^{n-i}\}\tilde{R}_F}}. \tag{21}$$

To maximize $R_{\text{ZFBF}}^{\text{multi}}$, the optimal caching thresholds $T_i^*, i = 0, \cdots, n-1$ can be found by enumeration of the solution space $0 \le T_i \le m,\ i = 0, \cdots, n-1,\ \sum_{i=0}^{n-1} T_i \le m$, which is of size m^n. While this complexity is exponential in n, it is not very large in practice since the number of caching thresholds is quite small ($n = \log_2 N$). For example, even for relatively large clusters with, say, 8 cooperating femto-BSs there are at most 3 thresholds.

5.2 Multi-antenna base stations

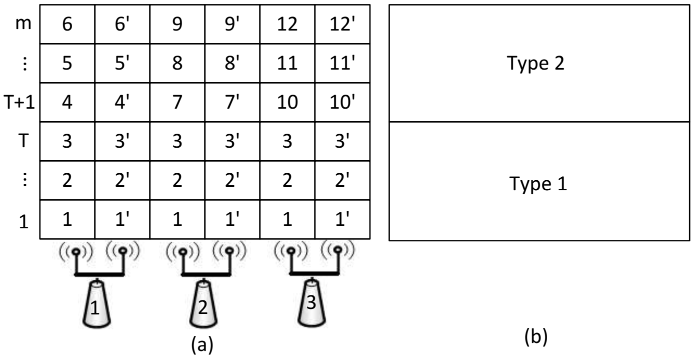


Fig. 5: Threshold-based caching and ZFBF with multi-antenna base stations ($m = 6,\ T = 3,\ N = 3,\ L_F = 2$).

Suppose that each femto-BS in the cluster has L_F antennas, the macro-BS has L_M antennas, and each user is still equipped with a single antenna. We consider the (single) threshold-based caching and ZFBF scheme in Section 4.2. That is, we first choose a threshold $T,\ 0 \le T \le m$. Then, we cache the files 1 to T in all of the femto-BSs, and cache the files $T+1$ to $T + N(m-T)$ in exactly one of the femto-BSs. Therefore, we have N copies for each of the most popular files 1 to T (referred to as type 1 files), one copy of the files $T+1$ to $T + N(m-T)$ (referred to as type 2 files), and the remaining files $T + N(m-T) + 1$ to M are stored in the macro-BS (referred to as type 3 files). As a result, the probability of a typical file request being type 1 is v_T. The probability of a typical file request being type 2 is $v_{T+N(m-T)} - v_T$. The probability of a typical file request being type 3 is $1 - v_{T+N(m-T)}$. Since each femto-BS has L_F antennas, it can potentially serve L_F users simultaneously with ZFBF (single-cell multi-user MIMO mode). For analytical purposes, we can think of each femto-BS having L_F "virtual" copies of its cache (see Fig. 5, e.g. file 1′ is a virtual version of file 1). This enables us to apply the analysis in Section 5.1 to the current case.

By using ZFBF in a multi-cell multi-user CoMP mode, where the ZFBF precoding is performed jointly across all antennas in all femto-BSs, we can simultaneously transmit $\min\{K, NL_F\}$ type 1 files with sum rate $\min\{K, NL_F\}\tilde{R}_F$. By using ZFBF in a single-cell multi-user MIMO mode, where ZFBF precoding is performed solely across the antennas in a single femto-BS, we can simultaneously transmit $\min\{K, L_F\}$ type 2 files with sum rate $\min\{K, L_F\}\tilde{R}_F$. By using ZFBF at the macro-BS (single-cell multi-user MIMO mode), we can simultaneously transmit $\min\{K, L_M\}$ type 3 files with sum rate $\min\{K, L_M\}\tilde{R}_M$. We divide an epoch into three portions with durations K_1, K_2, and K_3 for serving the type 1 file requests, type 2 file requests, and type 3 file requests respectively, where the epoch length is normalized to 1, i.e., $K_1 + K_2 + K_3 = 1$, like before. Following the same rational like before, the values of K_1, K_2, and K_3 satisfy the conservation law:

$$\begin{aligned} &K_1 \min\{K, NL_F\}\tilde{R}_F : K_2 \min\{K, L_F\}\tilde{R}_F \\ &\quad : K_3 \min\{K, L_M\}\tilde{R}_M \\ &\qquad = v_T : v_{T+N(m-T)} - v_T : 1 - v_{T+N(m-T)}. \qquad (22) \end{aligned}$$

The average effective data rate for the threshold-based caching and ZFBF scheme with multi-antenna BSs (denoted as $R^{\text{MIMO}}_{\text{ZFBF}}$) can be computed as

$$\begin{aligned} R^{\text{MIMO}}_{\text{ZFBF}}(T) &= K_1 \min\{K, NL_F\}\tilde{R}_F + K_2 \min\{K, L_F\}\tilde{R}_F \\ &\quad + K_3 \min\{K, L_M\}\tilde{R}_M \qquad (23) \\ &= \frac{1}{\frac{v_T}{\min\{K,NL_F\}\tilde{R}_F} + \frac{v_{T+N(m-T)}-v_T}{\min\{K,L_F\}\tilde{R}_F} + \frac{1-v_{T+N(m-T)}}{\min\{K,L_M\}\tilde{R}_M}}. \end{aligned}$$

Under the continuous approximation of the Zipf distribution, the optimal caching threshold is

$$T^* \approx \left[\frac{\xi N m}{1 + \xi(N-1)}\right]_0^m, \qquad (24)$$

where $\xi \triangleq \left(\frac{\frac{1}{\min\{K,NL_F\}\tilde{R}_F} - \frac{1}{\min\{K,L_F\}\tilde{R}_F}}{(N-1)\left(\frac{1}{\min\{K,L_F\}\tilde{R}_F} - \frac{1}{\min\{K,L_M\}\tilde{R}_M}\right)}\right)^{\frac{1}{s}}$.

5.3 Service time minimization

In Section 4, we characterize the system performance of the cache-driven femto-BS cooperation scheme in terms of the achieved rates. In this section we turn our attention to the service time and reformulate the analysis to minimize the average service time of a file request. We assume that the size of a file is L bits.

5.3.1 Randomized caching and MRT

Here our objective is to minimize the average service time of a file request (denoted as D_{MRT}). We have the following optimization problem:

$$\begin{aligned} \underset{q_1,\cdots,q_M}{\text{minimize}} \quad & \sum_{i=1}^{M} p_i U_D(q_i) \triangleq D_{\text{MRT}} \\ \text{subject to} \quad & \sum_{i=1}^{M} q_i \le m, \\ & 0 \le q_i \le 1, \ \forall i = 1, \cdots, M, \qquad (25) \end{aligned}$$

where $U_D(q_i)$ is the service time for the i-th file. Similar to Eq. (3), $U_D(q_i)$ can be computed as

$$U_D(q_i) = (1-q_i)^N \frac{L}{\tilde{R}_M} + \sum_{j=1}^{N} \binom{N}{j} q_i^j (1-q_i)^{N-j} \frac{L}{\tilde{R}_F^{(j)}}, \qquad (26)$$

where $\frac{L}{\tilde{R}_M}$ is the service time when the file request is served by the macro-BS and $\frac{L}{\tilde{R}_F^{(j)}}$ is the service time when the file request is served by a cluster of j femto-BSs. Following a similar procedure as in Section 4.1, we can find the conditions for the convexity of $U_D(q_i)$ and then solve for the optimal $q_i^*, \ i = 1, \cdots, M$ by the KKT conditions.

5.3.2 Threshold-based caching and ZFBF

Consider the (single) threshold-based caching and ZFBF scheme as in Section 4.2, where the probability of a typical file request being type 1 is v_T, the probability being type 2 is $v_{T+N(m-T)} - v_T$, and the probability being type 3 is $1 - v_{T+N(m-T)}$. Also, recall that when using ZFBF we can simultaneously serve $\min\{K, N\}$ type 1 files with sum rate $\min\{K, N\}\tilde{R}_F$. Type 2 (type 3) files are served one by one with rate $\tilde{R}_F$ ($\tilde{R}_M$). Then, if Q is the total number of file requests, it is easy to see that the average service time of a file request, denoted by D_{ZFBF}, can be computed as

$$\begin{aligned} D_{\text{ZFBF}}(T) &= \lim_{Q\to\infty} \frac{1}{Q}\Bigg[\frac{Q v_T L}{\min\{K,N\}\tilde{R}_F} \\ &\quad + \frac{Q(v_{T+N(m-T)} - v_T)L}{\tilde{R}_F} + \frac{Q(1 - v_{T+N(m-T)})L}{\tilde{R}_M}\Bigg] \\ &= L\left[\frac{v_T}{\min\{K,N\}\tilde{R}_F} + \frac{v_{T+N(m-T)} - v_T}{\tilde{R}_F} + \frac{1 - v_{T+N(m-T)}}{\tilde{R}_M}\right] \\ &= \frac{L}{R_{\text{ZFBF}}(T)}, \qquad (27) \end{aligned}$$

where $R_{\text{ZFBF}}(T)$ is the average effective data rate (see Eq. (11)). We can see that in the saturation regime, minimizing the service time $D_{\text{ZFBF}}(T)$ (with respect to the caching threshold T) is essentially the same as maximizing the rate $R_{\text{ZFBF}}(T)$.

6 Non-saturation analysis

In this section, we do not assume the saturation regime anymore. Now, file requests arrive over time and we use queueing theory to study the performance of the system. Specifically, we model the arrival of file requests as a Poisson process with rate λ and the identities of the file requests are drawn from the file popularity distribution in an i.i.d. manner. We compute the average system delay (including the queueing delay and the service time) of a file request.

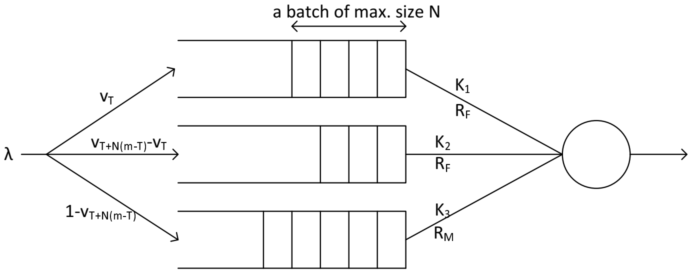


Fig. 6: Queueing model for threshold-based caching and ZFBF.

6.1 Randomized caching and MRT

The file requests arrive to the system according to a Poisson process with rate λ. The data rate for serving a file request depends on the number of femto-BSs caching the requested file, ranging in the set $\left\{\tilde{R}_M, \tilde{R}_F, \tilde{R}_F^{(2)}, \ldots, \tilde{R}_F^{(N)}\right\}$. Under the FIFO service discipline, the arrival and departure of the file requests can be modeled by an $M/G/1$ queue, where the arrival rate is λ and the service time is a random variable, denoted by X,

$$X = \begin{cases} \frac{L}{\tilde{R}_M} & \text{w/ prob.} \sum_{i=1}^{M} p_i (1-q_i)^N \\ \frac{L}{\tilde{R}_F^{(j)}} & \text{w/ prob.} \sum_{i=1}^{M} p_i \binom{N}{j} q_i^j (1-q_i)^{N-j}, j = 1, \ldots, N \end{cases} \tag{28}$$

Note that the average service time $\mathbb{E}[X]$ is what we called D_{MRT} in Section 5.3. Suppose $\lambda D_{\text{MRT}} < 1$. By the Pollaczek-Khinchine formula [26], the average system delay (including the queueing delay and the service time) of a file request, denoted by W_{MRT}, can be computed as

$$W_{\text{MRT}} = D_{\text{MRT}} + \frac{\lambda \mathbb{E}[X^2]}{2(1-\lambda D_{\text{MRT}})}. \tag{29}$$

6.2 Threshold-based caching and ZFBF

The file requests generated by different users arrive to the system according to a Poisson process with rate λ.

We organize the file requests according to their types into three queues. Type 1 requests, where the arrival rate is λv_T, are queued in queue 1 in order of their arrival. Similarly, type 2 requests, where the arrival rate is $\lambda(v_{T+N(m-T)} - v_T)$, are queued in queue 2 in order of their arrival. Type 3 requests, where the arrival rate is $\lambda(1 - v_{T+N(m-T)})$, are queued in queue 3 in order of their arrival.

We assume a single server and consider the weighted round robin service discipline with weights K_1, K_2, and K_3, where $K_1 + K_2 + K_3 = 1$ as discussed in Section 4.2. Specifically, we allocate K_3 portion of time to serve queue 3 with rate $\tilde{R}_M$ (the server is the macro-BS). We allocate K_2 portion of time to serve queue 2 with rate $\tilde{R}_F$ (the server is a femto-BS). We allocate K_1 portion of time to serve queue 1, where we collect a batch of up to N type 1 requests and serve them jointly with rate $\tilde{R}_F$ (the server is the femto-BS cooperation cluster of size N using ZFBF). Note that queue 1 is the so-called bulk queue with service in batches of maximum size N. See Fig. 6.

Before we compute the system delay, we consider the stability of the three queues. Queue 3 is essentially an $M/D/1$ queue with arrival rate $\lambda(1 - v_{T+N(m-T)})$ and service rate $K_3\tilde{R}_M$, which is stable if

$$\lambda < \frac{K_3\tilde{R}_M/L}{1 - v_{T+N(m-T)}} = \frac{R_{\text{ZFBF}}(T)}{L}, \tag{30}$$

where we use Eq. (6) and Eq. (7). Similarly, queue 2 is essentially an $M/D/1$ queue with arrival rate $\lambda(v_{T+N(m-T)} - v_T)$ and service rate $K_2\tilde{R}_F$, which is stable if

$$\lambda < \frac{K_2\tilde{R}_F/L}{v_{T+N(m-T)} - v_T} = \frac{R_{\text{ZFBF}}(T)}{L}. \tag{31}$$

Queue 1 is essentially an $M/D^{[1\cdots N]}/1$ bulk queue (with service in batches of maximum size N), where the arrival rate is λv_T and service rate is $K_1\tilde{R}_F$. Queue 1 is stable if

$$\lambda < \frac{K_1 N \tilde{R}_F/L}{v_T} = \frac{R_{\text{ZFBF}}(T)}{L}. \tag{32}$$

We can see that all three queues are stable when the arrival rate of file requests λ is less than the average effective service rate $R_{\text{ZFBF}}(T)/L$ (file requests per second), which is maximized when $T = T^*$ (see Eq. (9)).

The average system delay of a file request, denoted by W_{ZFBF}, can be computed as

$$W_{\text{ZFBF}} = v_T W_1 + (v_{T+N(m-T)} - v_T) W_2 + (1 - v_{T+N(m-T)}) W_3, \tag{33}$$

where $W_i,\ i = 1, 2, 3$ is the system delay for a type i request. By the Pollaczek-Khinchine formula for an $M/D/1$ queue with arrival rate $\lambda(1 - v_{T+N(m-T)})$ and service rate $K_3\tilde{R}_M$, W_3 can be obtained as

$$W_3 = \frac{(2 - \frac{\lambda L}{R_{\text{ZFBF}}(T)})L}{2(1 - \frac{\lambda L}{R_{\text{ZFBF}}(T)}) R_{\text{ZFBF}}(T)(1 - v_{T+N(m-T)})}. \tag{34}$$

Similarly, for an $M/D/1$ queue with arrival rate $\lambda(v_{T+N(m-T)} - v_T)$ and service rate $K_2\tilde{R}_F$, W_2 can be obtained as

$$W_2 = \frac{(2 - \frac{\lambda L}{R_{\text{ZFBF}}(T)})L}{2(1 - \frac{\lambda L}{R_{\text{ZFBF}}(T)}) R_{\text{ZFBF}}(T)(v_{T+N(m-T)} - v_T)}. \tag{35}$$

The system delay for an $M/D^{[1\cdots N]}/1$ bulk queue does not have a closed-form formula. We resort to numerical methods to obtain W_1 (see Section 7).

Under medium/low loads, the asynchronous arrival of file requests poses the dilemma to wait till enough file requests are collected to concurrently serve many users (resulting in a larger multiplexing gain) or to immediately serve less users (resulting in a smaller multiplexing gain). Specifically, when the size of queue 1 is less than N, we may proceed to serve queue 2 (or queue 3) until enough type 1 requests are accumulated.

We propose the following policy to address the above dilemma and to "fully" exploit the multiplexing gain. Recall that the file requests are organized into three queues, one for each of the three type requests. In addition, there is a (weighted round robin) selector that decides on which queue the server will work (and the corresponding time portion). Suppose that the selector checks queue 1. When the number of the type 1 requests is larger than or equal to N, they are served in batches of size N for a portion of time

K_1. Otherwise, when the number of the type 1 requests is smaller than N, we have the following two cases. The first case is that there are pending type 2 or type 3 requests, under which we skip the round for serving queue 1 and proceed to serve type 2 requests or type 3 requests until N type 1 requests are accumulated. The second case is that both queues for the type 2 and type 3 requests are empty, under which the type 1 requests are served in a batch (of size less than N).

Note that under the above policy, the type 1 file requests are served in a batch of size less than N only if the queues for the type 2 and type 3 requests are empty. In this way, we fully exploit the gains from the spatial multiplexing.

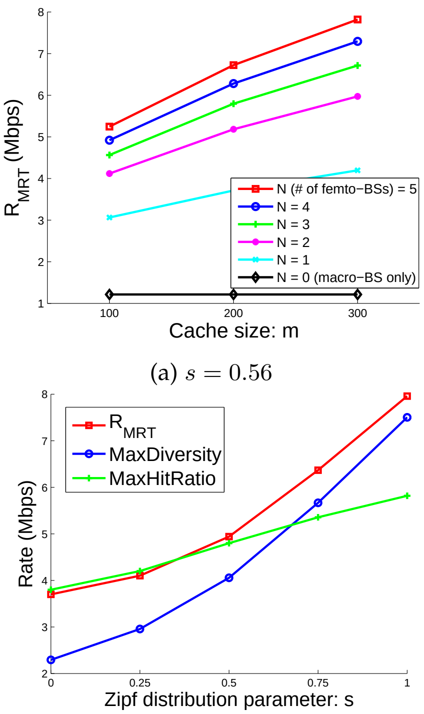
7 Numerical results

In this section we present performance results by numerically solving our analytical model in a number of practical scenarios. We assume that there is a library of $M = 1000$ video files, and the file popularity distribution follows the Zipf distribution [4], [27] with parameter s, i.e., $p_i = c_{M,s}/i^s$, $i = 1, \ldots, M$, where $c_{M,s}$ is the normalization constant.

Without loss of generality, consider a macro-BS with transmission range 4000m and a number of femto-BSs each with transmission range 200m deployed inside the macro-BS cell. Note that these are typical transmission ranges for a macrocell and for low power BSs, see, for example, the capabilities of picocells, microcells, and femtocells defined in [28]. The number of femto-BSs inside a cluster (N), that is, the number of femto-BSs that a typical user can receive useful signal from, depends on the density of the femto-BSs and varies in the scenarios that we study. The data rate of the macro-BS is assumed to be $R_M = 2$ Mbps and the data rate of the femto-BSs is assumed to be $R_F = 10$ Mbps, again in line with industry practice. The transmit power of the macro-BS equals $P_M = 20$ W and of the femto-BSs equals $P_F = 20$ mW, as has been assumed in prior works as well [1].

Without loss of generality, consider a cluster of femto-BSs covering an area of radius 200m at distance 2000m from the macro-BS. Then, the distance of the typical user from a femto-BS of the cluster is assumed to lie between 0 and 200m, and the distance from the macro-BS lies between 1800m and 2200m. We assume that the path loss exponent equals $\alpha = 4$ and consider quasi-static Rayleigh flat-fading channels with unit mean power. In addition, the bandwidth of the frequency slot is $W = 5$ MHz and the noise power spectral density varies from $N_0 = 4 \times 10^{-19}$ W/Hz to $N_0 = 8 \times 10^{-20}$ W/Hz. As a result, by substituting these numerical values into the formulas of Section 4, the transmission success (non-outage) probability for the macro- and the femto-BSs with a unit diversity varies from 0.6 to 0.9, and the effective data rate of the macro- and femto-BSs with a unit diversity varies from $\tilde{R}_M = 2 \times 0.6 = 1.2$ to 1.8 Mbps and from $\tilde{R}_F = 10 \times 0.6 = 6$ to 9 Mbps, respectively.

7.1 Data rates under diversity gains



(a) $s = 0.56$

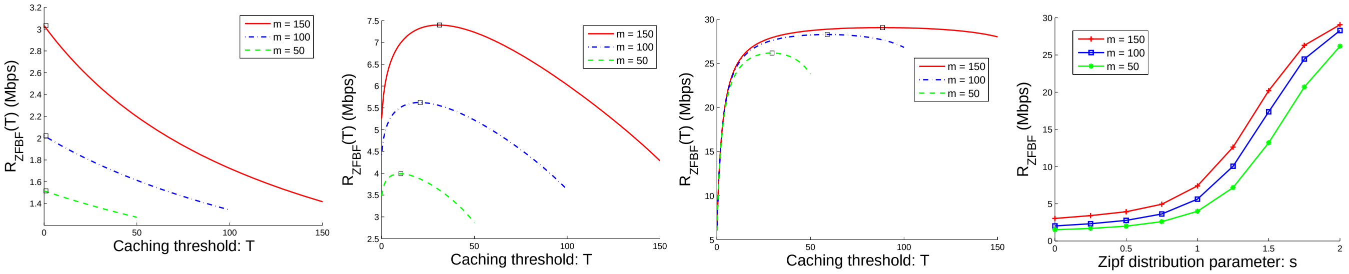
(b) $N = 5,\ m = 100$

Fig. 7: Performance of the system under MRT.

We study the rate of a user under MRT achieved at an arbitrary time slot. To highlight the effect of diversity gains to the data rate, we show results when the success (non-outage) probability with a unit diversity equals 0.6. We later show results when the outage probability is smaller.

In Fig. 7a we plot the rate as a function of the cache size (m) and the number of nearby femto-BSs (N) for a popularity distribution with a typical parameter, say, $s = 0.56$. As expected, the rate R_{MRT} increases with the cache size and the number of neighboring femto-BSs, the later because as N increases we have a larger diversity that reduces the transmission link failure (outage) probability. In this plot we also show the achieved rate when no femto-BSs are used. It is evident that using femto-BSs improves rates by 2-3x and adding MRT results in an additional gain of 2-3x. Last, note that for a fixed cache size and as we increase the number of femto-BSs, the marginal gain decreases since the outage probability has already been reduced to a very small value.

In Fig. 7b we plot the rate as a function of the Zipf distribution parameter s. We vary s from 0 (uniform distribution) to 1 (skewed distribution) and compare our scheme with the following two basic schemes: a "MaxDiversity" caching scheme and a "MaxHitRatio" caching scheme, to investigate how well our system adapts to changing levels of popularity. In the "MaxDiversity" scheme, we cache the most popular files 1 to m in every femto-BS so that we have a diversity of order N for all these files. Intuitively, this scheme would perform well for a skewed distribution with a large s. On the other hand, in the "MaxHitRatio" scheme we cache in the femto-BSs the most popular files 1 to mN, each with a single copy. This scheme would perform well for a near-flat (uniform) popularity distribution with a small s. As shown in Fig. 7b, our proposed cross-layer optimization scheme adapts to the popularity distribution and controls the diversity gain (equivalently, the number of copies of a



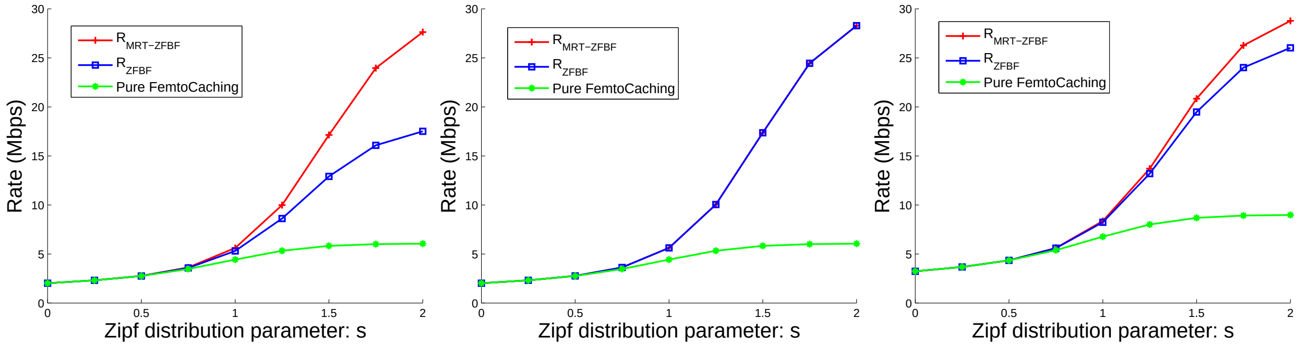
(a) $s = 0$ (uniform popularity distribution)

(b) $s = 1$ (skewed)

(c) $s = 2$ (very skewed)

(d) $s \in [0, 2]$

Fig. 8: Performance of the system under ZFBF, $N = 5$.



(a) $K = 3$

(b) $K \geq 5$

(c) $K = 3$, low outage probability of 0.1

Fig. 9: Performance of the system under pure FemtoCaching, ZFBF, and MRT–ZFBF, $N = 5$, $m = 100$.

file in the caches of the femto-BSs) for each individual file, attending a good performance in the whole range of s.

7.2 Data rates under multiplexing gains

Fig. 8a-c plots the rate $R_{\text{ZFBF}}(T)$ as a function of the caching threshold T, $0 \leq T \leq m$. We assume that the cluster of neighboring femto-BSs consists of $N = 5$ femto-BSs. We consider three values for the cache size, $m = 50$, 100 and 150, and three values for the Zipf distribution parameter, $s = 0$, 1 and 2. In practice, a cluster of $N = 5$ femto-BSs may serve tens to hundreds of users. In other words, we assume $K \geq N$.

From Fig. 8a we observe that for a uniform popularity distribution ($s = 0$) the optimal threshold is $T^* = 0$, i.e., we prefer to have only a single copy of a file in the caches to increase cache hits and choose not to have a multiplexing gain. On the other hand, in Fig. 8c, for the very skewed popularity distribution ($s = 2$), the optimal threshold T^* is closer to m, i.e., we prefer to have multiple copies of the most popular files in the caches of femto-BSs, achieving a large multiplexing gain. Note that for the very skewed distribution, we have $v_m \approx 1$ (v_m is the probability that a user requests one of the m most popular files) and $R_{\text{ZFBF}}(m) \approx N\tilde{R}_F$, where the multiplexing gain is N. For a skewed distribution, as shown in Fig. 8b with $s = 1$, the optimal threshold lies somewhere in the middle. Thus, our caching scheme can adapt to the file popularity distribution by properly setting the value of the caching threshold. Last, note that the rate gain from using ZFBF on the femto-BSs can be very substantial, reaching 10-20x in the case of very skewed distributions. Fig. 8a-c also show the increase in rate with respect to the cache size m. We observe that the optimal caching threshold T^* is proportional to m, as suggested by Eq. (9). Finally, in Fig. 8d, we plot the achieved rate (at the optimal threshold T^*) for the entire range of s and m.

7.3 Data rates under joint MRT–ZFBF

Fig. 9 compares the rates for “pure FemtoCaching”, ZFBF (R_{ZFBF}), and MRT–ZFBF ($R_{\text{MRT–ZFBF}}$), where in the latter two cases the rates are evaluated at their optimal caching thresholds, respectively. In the “pure FemtoCaching” scheme, we cache in the femto-BSs the most popular files 1 to mN, each with a single copy, and there is no cooperative transmission among the femto-BSs.

As shown in Fig. 9a, the MRT–ZFBF scheme outperforms the ZFBF scheme when the number of users K is less than the number of femto-BSs N. The reason is that if $K < N$, MRT–ZFBF achieves a diversity of order $N - K + 1 \geq 2$ for all the K users, increasing the transmission success (non-outage) probability from 0.6 to more than 0.9 (equality for a diversity of 2), which, in turn, increases the effective data rate. Of course, when $K \geq N$, ZFBF performs the same as MRT–ZFBF, as shwon in Fig. 9b.

The results above are generated when the success (non-outage) probability with a unit diversity equals 0.6. For a larger success probability, e.g. 0.9, the ZFBF scheme is expected to be very close to MRT–ZFBF. Fig. 9c confirms the above. Last, we can see that our proposed caching strategies outperform the pure FemtoCaching scheme, especially for a skewed file popularity.

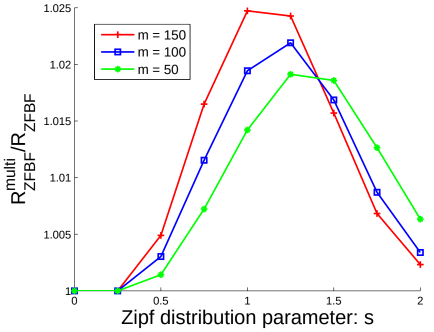


Fig. 10: Performance of the system under ZFBF with multiple thresholds, $N = 4$.

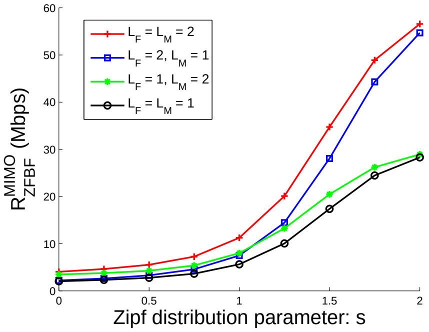


Fig. 11: Performance of the system under ZFBF with multi-antenna BSs, $N = 5$, $m = 100$.

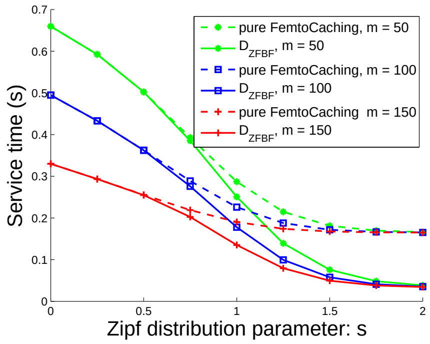


Fig. 12: Performance of the system in terms of service time, $N = 5$.

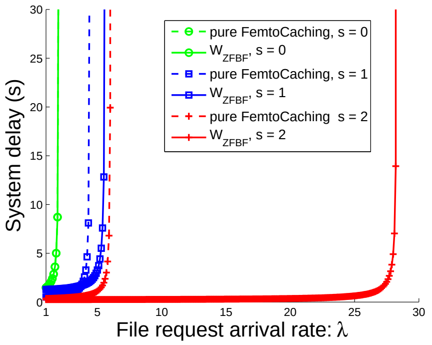


Fig. 13: Performance of the system in terms of system delay, $N = 5$, $m = 100$.

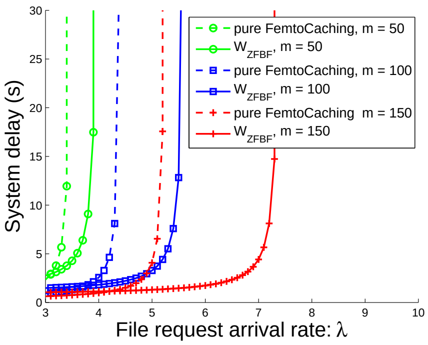


Fig. 14: Performance of the system in terms of system delay, $N = 5$, $s = 1$.

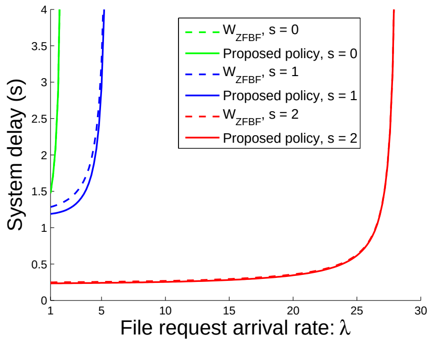


Fig. 15: Performance of the proposed policy concerning the asynchronous arrival of file requests, $N = 5$, $m = 100$.

7.4 Single threshold vs. multiple thresholds

Fig. 10 plots the rate when two thresholds are used ($R^{\text{multi}}_{\text{ZFBF}}(T_0^*, T_1^*)$), over the rate when one threshold is used ($R_{\text{ZFBF}}(T^*)$) in a setup with 4 femto-BSs ($N = 2^n = 4$). We can see that $R^{\text{multi}}_{\text{ZFBF}}$ is at most 2.5% larger than $R_{\text{ZFBF}}(T^*)$ over the whole range of file popularity distributions (s) and cache size (m). Thus, the use of a single threshold appears to be enough to get most of the rate gains.

7.5 Multi-antenna base stations

Fig. 11 plots the rate $R^{\text{MIMO}}_{\text{ZFBF}}$ as a function of the Zipf distribution parameter s under different combinations of the number of antennas at femto-BSs and the macro-BS. We consider a cluster of $N = 5$ femto-BSs with $L_F = 1$ or 2 antennas and a macro-BS with $L_M = 1$ or 2 antennas. We can see that the rate is significantly increased by having an additional antenna at femto-BSs to provide spatial multiplexing, especially when the file popularity is skewed ($s > 1$). On the other hand, when s is small (near flat popularity), it is slightly better to have an additional antenna at the macro-BS (single-cell multi-user MIMO) than at femto-BSs.

7.6 Service time

In Fig. 12, we compare the performance of the threshold-based caching and ZFBF scheme (D_{ZFBF}) with the pure FemtoCaching scheme in terms of the average service time of serving a file request with file size $L = 1$Mb. We can see that the ZFBF scheme outperforms pure FemtoCaching, especially when s is large. Note that when $s = 2$, the performance of both the pure FemtoCaching and the ZFBF scheme are insensitive to the cache size since the file popularity is concentrated on a small fraction (< 50) of files.

7.7 Non-saturation analysis

We show results that take into account the fact that file requests will arrive with some rate and the system is not necessarily in the saturation regime. In Fig. 13, we compare the performance of the threshold-based caching and ZFBF scheme (W_{ZFBF}) with the pure FemtoCaching scheme in terms of the average system delay under a Poisson arrival of file requests with rate $\lambda = 1$ to 30 requests per second. The file size is set as $L = 1$Mb. As the arrival rate λ increases, the system delay diverges, starting from smaller values of s to larger values of s. Also, the system can be stabilized for a larger range of arrival rates with the ZFBF scheme when compared to the stability region for the pure FemtoCaching (especially when s is large). The reason is that for a skewed file popularity, in the ZFBF scheme we cache multiple copies of the most popular files in the caches to achieve a larger multiplexing gain, and, in turn, we achieve a larger service rate. In Fig. 14, similar results can be observed as we fix the value of $s = 1$ and vary the cache size $m = 50$, 100, and 150.

Last, Fig. 15 shows the performance of the proposed policy in Section 6.2 which attempts to maximize the multiplexing gains by deferring serving type 1 requests up until there are many such requests in the queue that can be served concurrently. We can see that the proposed policy results in a better performance than the original weighted round robin policy. This is because the proposed policy achieves higher multiplexing gains on average. When saturation occurs, the queues are quite long, and, as a result, the original weighted round robin policy also achieves high multiplexing gains and the performance of the two schemes is almost the same.

8 Practical considerations

Updating the parameters of the caching strategy: The parameters of the caching strategy depend on the file popularity distribution p_i, e.g. q_i in Eq. (4) for MRT and the threshold T in Eq. (9) for ZFBF. The file popularity distribution can be estimated by file requests, see, for example, [29]. The complexity of computing the caching strategy parameter q_i in randomized caching under MRT is low since q_i can be obtained efficiently by solving a convex optimization problem, see Theorem 1. The complexity of computing T in threshold-based caching is also tractable since the size of the solution space is linear in the cache size for the single threshold case. In addition, when the discrete file popularity distribution can be approximated by a continuous one, the optimal threshold can be obtained analytically. For the multi-threshold case the solution space is exponential in the number of thresholds, but (i) even for large cluster sizes, e.g. 8 femto-BSs, the number of thresholds is as small as 3, and (ii) as we have shown in Section 7.4 the use of a single threshold is enough to get most of the rate gains thus we do not anticipate the use of multiple thresholds in the majority of cases. The right place to perform the above computations is the macro-BS as it can keep track of all file requests and thus of the file popularity distribution, and it has more than enough computation power to compute the caching strategy parameters. Upon computing the parameters, the macro-BS will send their values to the femto-BSs. Last, since the time scale of significant changes in the file popularity distribution is in the order of a day or longer, updating the caching strategy parameters will happen infrequently.

Cache content update: Macro-BS and femto-BSs coordinate the cache content update (downloading popular video files via backhaul into caches in femto-BSs) according to updates in the caching strategy parameters. This can be done at off-peak hours because the time scale of significant changes in the file popularity distribution (e.g. days) is much larger than the time scale of receiving users' requests (e.g. seconds) [3]. Note also that only significant changes in the file popularity will result in a cache content update, while small changes in the file popularity will only result in reordering (relabeling) files in the caches.

9 Conclusion

In this paper we proposed a new system architecture that jointly uses and optimizes distributed caching in femto-BSs and femto-BS cooperative transmissions. Our analytical and simulation results show that our system achieves an order of magnitude faster content delivery than legacy systems. The gains are particularly pronounced for skewed popularity distributions where caching multiple copies of popular files across multiple femto-BSs yields particularly large diversity and multiplexing gains without sizeably increasing cache misses. Given that content popularity is well known to be heavily skewed, our approach is expected to have a large impact in real-world setups.

References

[1] V. Chandrasekhar, J. G. Andrews, and A. Gatherer, "Femtocell networks: a survey," *IEEE Commun. Mag.*, vol. 46, no. 9, pp. 59–67, Sep. 2008.

[2] A. Ghosh, N. Mangalvedhe, R. Ratasuk, B. Mondal, M. Cudak, E. Visotsky, T. A. Thomas, J. G. Andrews, P. Xia, H. S. Jo, H. S. Dhillon, and T. D. Novlan, "Heterogeneous cellular networks: From theory to practice," *IEEE Commun. Mag.*, vol. 50, no. 6, pp. 54–64, Jun. 2012.

[3] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *Proc. IEEE INFOCOM*, 2012.

[4] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.

[5] 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, Apr. 2013.

[6] R. Irmer, H. Droste, P. Marsch, M. Grieger, G. Fettweis, S. Brueck, H. P. Mayer, L. Thiele, and V. Jungnickel, "Coordinated multipoint: Concepts, performance, and field trial results," *IEEE Commun. Mag.*, vol. 49, no. 2, pp. 102–111, Feb. 2011.

[7] D. Lee, H. Seo, B. Clerckx, E. Hardouin, D. Mazzarese, S. Nagata, and K. Sayana, "Coordinated multipoint transmission and reception in LTE-advanced: deployment scenarios and operational challenges," *IEEE Commun. Mag.*, vol. 50, no. 2, pp. 148–155, Feb. 2012.

[8] H. V. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, and G. Caire, "Achieving high data rates in a distributed MIMO system," in *Proc. ACM MOBICOM*, 2012.

[9] H. S. Rahul, S. Kumar, and D. Katabi, "JMB: Scaling wireless capacity with user demands," in *Proc. ACM SIGCOMM*, 2012.

[10] H. V. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, and G. Caire, "AirSync: Enabling distributed multiuser MIMO with full spatial multiplexing," *IEEE/ACM Trans. Netw.*, vol. 21, no. 6, pp. 1681–1695, Dec. 2013.

[11] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[12] D. Ben Cheikh, J. M. Kelif, M. Coupechoux, and P. Godlewski, "Analytical joint processing multi-point cooperation performance in Rayleigh fading," *IEEE Wireless Commun. Lett.*, vol. 1, no. 4, pp. 272–275, Aug. 2012.

[13] D. Gesbert, S. Hanly, H. Huang, S. Shitz, O. Simeone, and W. Yu, "Multi-cell MIMO cooperative networks: A new look at interference," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 9, pp. 1380–1408, Dec. 2010.

[14] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of distributed caching in D2D wireless networks," in *Proc. IEEE ITW*, 2013.

[15] N. Golrezaei, A. Dimakis, and A. Molisch, "Scaling behavior for device-to-device communications with distributed caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 4286–4298, Jul. 2014.

[16] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[17] ——, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1029–1040, Aug. 2015.

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

[19] M. Haenggi, J. G. Andrews, F. Baccelli, O. Dousse, and M. Franceschetti, "Stochastic geometry and random graphs for the analysis and design of wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 7, pp. 1029–1046, Sep. 2009.

[20] H. S. Dhillon, R. K. Ganti, F. Baccelli, and J. G. Andrews, "Modeling and analysis of K-tier downlink heterogeneous cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 550–560, Apr. 2012.

[21] A. Liu and V. K. N. Lau, "Mixed-timescale precoding and cache control in cached MIMO interference network," *IEEE Trans. Signal Process.*, vol. 61, no. 24, pp. 6320–6332, Dec. 2013.

[22] ——, "Cache-enabled opportunistic cooperative MIMO for video streaming in wireless systems," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 390–402, Jan. 2014.

[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[24] K.-K. Wong and Z. Pan, "Array gain and diversity order of multiuser MISO antenna systems," *Int. J. Wireless Inf. Networks*, 2008.

[25] M. Haenggi and R. K. Ganti, "Interference in large wireless networks," *Found. Trends Netw.*, vol. 3, no. 2, pp. 127–248, Feb. 2009.

[26] D. Bertsekas and R. Gallager, *Data Networks*. Prentice-Hall, Inc., 1992.

[27] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of YouTube network traffic at a campus network - measurements, models, and implications," *Comput. Netw.*, vol. 53, no. 4, pp. 501–514, Mar. 2009.

[28] http://www.smallcellforum.org/.

[29] F. Figueiredo, F. Benevenuto, and J. M. Almeida, "The tube over time: characterizing popularity growth of YouTube videos," in *Proc. ACM WSDM*, 2011.

Appendix

Proof of Theorem 1:

First, let us define the function

$$G(q) \triangleq \sum_{j=0}^{N} \binom{N}{j} q^j (1-q)^{N-j} w_j, \; 0 \le q \le 1, \tag{36}$$

where $w_j \ge 0, \; \forall j = 0, \cdots, N$. We have the following lemma.

Lemma 1. *If $w_j \ge w_{j-1}, \; \forall j = 1, \cdots, N$, then $G(q)$ is a non-decreasing function. Furthermore, if $w_{j+1} - w_j \le w_j - w_{j-1}, \; \forall j = 1, \cdots, N-1$, then $G(q)$ is concave.*

Proof. Using basic analysis, the first derivative of $G(q)$ can be computed as

$$\begin{aligned}
G'(q) &= \sum_{j=1}^{N} \binom{N}{j} j q^{j-1} (1-q)^{N-j} w_j \\
&- \sum_{j=0}^{N-1} \binom{N}{j} (N-j) q^{j} (1-q)^{N-j-1} w_j \\
&= \sum_{j=1}^{N} \binom{N}{j} j q^{j-1} (1-q)^{N-j} w_j \\
&- \sum_{j=1}^{N} \binom{N}{j-1} (N-j+1) q^{j-1} (1-q)^{N-j} w_{j-1} \\
&= \sum_{j=1}^{N} \frac{N!}{(j-1)!(N-j)!} q^{j-1} (1-q)^{N-j} (w_j - w_{j-1}).
\end{aligned} \tag{37}$$

When $w_j \ge w_{j-1}, \forall j = 1, \cdots, N$, we have $G'(q) \ge 0$. So, $G(q)$ is non-decreasing.

Similarly, the second derivative of $G(q)$ is computed as

$$\begin{aligned}
G''(q) &= \sum_{j=2}^{N} \binom{N}{j} j(j-1) q^{j-2} (1-q)^{N-j} w_j \\
&- \sum_{j=1}^{N-1} \binom{N}{j} j(N-j) q^{j-1} (1-q)^{N-j-1} w_j \\
&- \sum_{j=1}^{N-1} \binom{N}{j} j(N-j) q^{j-1} (1-q)^{N-j-1} w_j \\
&+ \sum_{j=0}^{N-2} \binom{N}{j} (N-j)(N-j-1) q^{j} (1-q)^{N-j-2} w_j \\
&= \sum_{j=1}^{N-1} \binom{N}{j+1} (j+1) j q^{j-1} (1-q)^{N-j-1} w_{j+1} \\
&- \sum_{j=1}^{N-1} \binom{N}{j} j(N-j) q^{j-1} (1-q)^{N-j-1} w_j \\
&- \sum_{j=1}^{N-1} \binom{N}{j} j(N-j) q^{j-1} (1-q)^{N-j-1} w_j \\
&+ \sum_{j=1}^{N-1} \binom{N}{j-1} (N-j+1)(N-j) q^{j-1} (1-q)^{N-j-1} w_{j-1} \\
&= \sum_{j=1}^{N-1} \frac{N!}{(j-1)!(N-j-1)!} q^{j-1} (1-q)^{N-j-1} \\
&\cdot \left[(w_{j+1} - w_j) - (w_j - w_{j-1}) \right].
\end{aligned} \tag{38}$$

When $w_{j+1} - w_j \le w_j - w_{j-1}, \; \forall j = 1, \cdots, N-1$, we have $G''(q) \le 0$. So, $G(q)$ is concave. □

Let w_0 represent the effective data rate of a macro-BS ($w_0 = \tilde{R}_M$) and w_j represent the effective data rate of a femto-BS cluster with a diversity of order j ($w_j = \tilde{R}_F^{(j)}$), $j = 1, \cdots, N$. From Eq. (3) and Eq. (36), we have $U(q_i) = \sum_{j=0}^{N} \binom{N}{j} q_i^j (1-q_i)^{N-j} w_j = G(q_i)$. In addition, since the ccdf of $S_{\Gamma(j)}$ is $\overline{F}_{S_{\Gamma(j)}}(z) = \sum_{n=0}^{j-1} \frac{1}{n!} e^{-z} z^n$, we obtain

$$\begin{aligned}
w_{j+1} - w_j &= R_F \sum_{n=0}^{j} \frac{1}{n!} e^{-\eta_F} \eta_F^n - R_F \sum_{n=0}^{j-1} \frac{1}{n!} e^{-\eta_F} \eta_F^n \\
&= R_F \frac{1}{j!} e^{-\eta_F} \eta_F^j, \; j = 1, \cdots, N-1;
\end{aligned} \tag{39}$$

$$w_1 - w_0 = R_F e^{-\eta_F} - R_M e^{-\eta_M}. \tag{40}$$

We note that the conditions $w_{j+1} \ge w_j, \; j \ge 0$ are equivalent to $R_F e^{-\eta_F} \ge R_M e^{-\eta_M}$. Also, the conditions $w_{j+1} - w_j \le w_j - w_{j-1}, \; j \ge 2$ (the marginal rate gain of including one more femto-BS into the cluster to perform cooperative transmission is decreasing) are equivalent to $\eta_F \le 2$. Moreover, the condition $w_2 - w_1 \le w_1 - w_0$ (the aforementioned marginal gain is smaller than the difference between the rates of a femto-BS and a macro-BS) is equivalent to $R_M e^{-\eta_M} \le R_F e^{-\eta_F}(1 - \eta_F)$. As a result, by Lemma 1 and combining the above conditions, we conclude that if $R_M e^{-\eta_M} \le R_F e^{-\eta_F}(1 - \eta_F)$ holds, $U(q_i)$ is concave and thus $R_{\text{MRT}}(q_1, \cdots, q_M) = \sum_{i=1}^{M} p_i U(q_i)$ is concave.