SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

August 2019, Vol. 62 082301:1–082301:14
https://doi.org/10.1007/s11432-018-9680-9

Delay-constrained sleeping mechanism for energy saving in cache-aided ultra-dense network

Pei LI[1], Shulei GONG[2,3], Shen GAO[1], Yaoyue HU[1], Zhiwen PAN[1*] & Xiaohu YOU[1]

[1]*National Mobile Communications Research Laboratory, Southeast University, Nanjing* 210096*, China;*
[2]*Institute of Space-Terrestrial Intelligent Networks, School of Electronic Science and Engineering, Nanjing University, Nanjing* 210093*, China;*
[3]*China Mobile Group Jiangsu Co., Ltd., Nanjing* 210029*, China*

Received 15 July 2018/Revised 26 September 2018/Accepted 29 November 2018/Published online 28 May 2019

Abstract We investigate an energy-saving sleeping mechanism in a cache-aided ultra-dense network (UDN) with delay constraints. As in existing works, we consider the video and file contents of the UDN. The video contents are cached at and delivered by a small-cell base-station (sBS). The cache-aided sBS cooperates with a macro-cell base-station (mBS) to service the file contents. The optimal sleeping strategy that conserves energy under the delay constraint is formulated as an energy-consumption minimization problem under the network stability condition with a guaranteed delay constraint. To find its solution, the minimization problem is transformed into a joint optimization problem of energy consumption and delay by the Lyapunov technique. A delay-constrained sleeping algorithm is proposed, and its effectiveness is confirmed by the numerical results of a simulation study. A tradeoff between energy consumption and delay, achieved by adjusting the weighting factor in the cache-aided UDN, is also demonstrated.

Keywords ultra-dense networks, sleeping mechanism, mean delay, caching strategy, energy saving

Citation Li P, Gong S L, Gao S, et al. Delay-constrained sleeping mechanism for energy saving in cache-aided ultra-dense network. Sci China Inf Sci, 2019, 62(8): 082301, https://doi.org/10.1007/s11432-018-9680-9

1 Introduction

The dense deployment of small cell base-stations (sBSs) in an ultra-dense network (UDN) improves the throughput [1], but consumes a large amount of energy. The energy consumption in a UDN can be reduced by a BS sleeping scheme that turns off the low-load BSs [2]; however, this proposal increases the delay. Hence, an efficient sleeping scheme should not sacrifice the delay performance of the system.

The sleeping mechanism under delay constraints has been extensively researched. A random sleeping strategy and a traffic-aware sleeping strategy reportedly maximize the energy efficiency under a delay constraint [3]. Liu et al. [4] developed a random sleeping strategy based on the N-policy M/G/1 queueing model, in which each BS independently enters the sleeping state without considering other BS states. This scheme saves energy while satisfying the delay requirement. A BS sleeping strategy that extends the sleeping period when the UE can tolerate a delay was proposed in [5].

Obviously, there exists a tradeoff between energy saving and delay. The authors of [6] devised an energy-saving, greedy-on greedy-off BS sleeping strategy with a flexible energy-delay tradeoff. In [7], the energy-efficiency vs. delay tradeoff in virtualized wireless networks was formulated as an energy efficiency maximization problem that constrains the user rate requests and delay limit. In our previous work [8], we

* Corresponding author (email: pzw@seu.edu.cn)

formulated the energy-delay tradeoff problem as a cost minimization problem, and optimized the sleeping time of the BSs.

BSs have recently been equipped with storage capabilities, imbuing them with a caching policy [9]. Cache-aiding can considerably reduce the energy consumption and delay in a heterogeneous network. To maximize the caching performance and energy efficiency, Chen et al. [10] combined cooperative caching with a transmission policy for cluster-centric networks of cache-enabled small cells. Xu et al. [11] applied two-edge in-memory caching policies at different times. Their strategy improves the energy efficiency of three-tier heterogeneous networks. In [12], the power consumptions of caching and uplink were jointly minimized by an energy-efficient content-placement policy developed through integer linear programming.

More recently, caching policies have been combined with sleeping strategies for further conservation of energy. Poularakis et al. [13] combined a caching and BS activation strategy into an approximation framework that transforms the energy minimization problem into a maximization problem of the submodular function under knapsack constraints. Xie et al. [14] iteratively solved the joint caching-BS activation optimization problem through a quantum-inspired evolutionary algorithm. In [15], the energy saving was maximized by a caching algorithm based on dynamic energy harvesting with a sleep-active scheduling mechanism.

However, in previous works, the caching and BS activation have been co-optimized for energy saving alone, without considering the delay. In this paper, we study a delay-constrained BS sleeping scheme that conserves energy in a cache-aided UDN.

As implemented in related papers [16, 17], our strategy considers the video and file contents requested by users. The video contents are cached at a small-cell base station (sBS) and delivered by the sBS. To service the file contents, the cache-aided sBS cooperates with a macro-cell base-station (mBS).

The main contributions of this paper are summarized below:

• We obtain the transmission probabilities, which depend on the caching probability and state of the sBS, in an M/G/1 processor-shared queueing model.

• We express the average energy consumption in the system, derive the mean delay in delivering the file and video contents, and analyze the impacts of the sleeping scheme and its cache capacity.

• We reformulate the delay-constrained sleeping strategy that maximizes the energy saving as an energy minimization problem constrained by the network stability (a proxy of the delay). The problem is solved by a delay-constrained sleeping algorithm based on the Lyapunov optimization theory.

The remainder of this paper is organized as follows. Section 2 introduces the system model and its operation mode. Section 3 analyzes the queueing model and mean delay in the network, and Section 4 investigates the delay-constrained sleeping mechanism for the energy-saving problem. Numerical and simulation results are provided in Section 5, and conclusion is presented in Section 6.

2 System model and operation modes

2.1 Network model

Consider the downlink of a two-tier UDN consisting of N_s sBSs overlaid by N_m mBSs. Assume that the location distributions of the sBSs and mBSs follow an independent Poisson point process (PPP) with intensities λ_s and λ_m ($\lambda_s > \lambda_m$) respectively [8], as shown in Figure 1.

To conserve energy, a sleeping strategy is imposed on the sBSs. While the mBSs are permanently active, the sBSs are either active or sleeping. In the sleeping state, the transceivers and other hardware components of the sBS are turned off, blocking the transmission service of that sBs. We denote the BS state by $\boldsymbol{S} = (s_1, s_2, \ldots s_k, \ldots, s_{N_s})$, where $s_k \in \{0, 1\}$. The sleeping and active states of sBS k are represented by $s_k = 0$ and $s_k = 1$, respectively. The sets of sleeping and active sBS are denoted by $\mathcal{B}_{\text{on}}$ and $\mathcal{B}_{\text{off}}$, respectively, and the total number of active and sleeping sBS are expressed as $|\mathcal{B}_{\text{on}}|$ and $|\mathcal{B}_{\text{off}}|$, respectively.

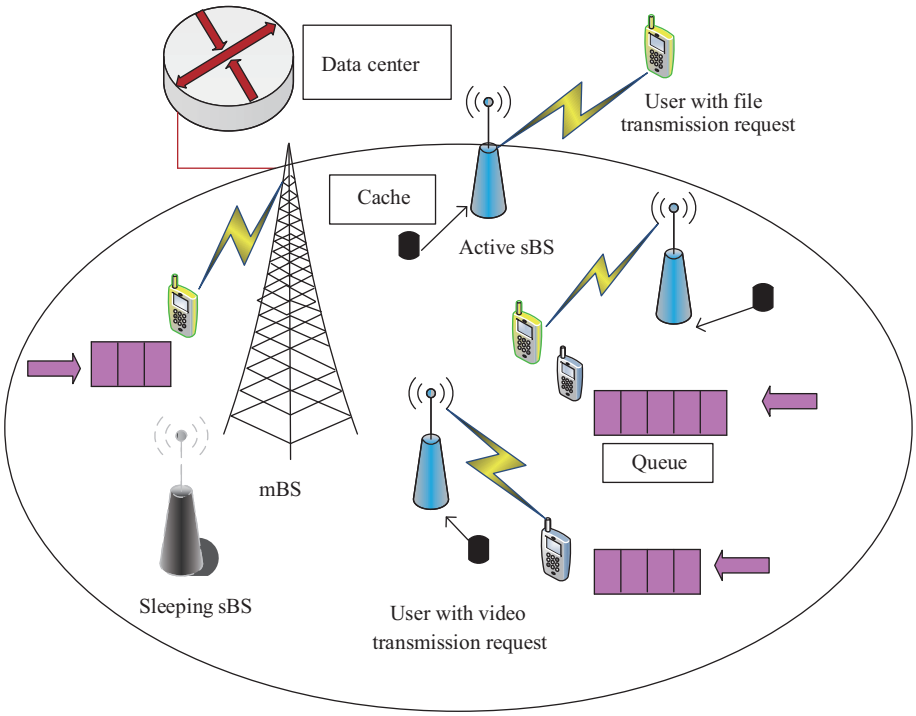

Figure 1 (Color online) System model for a cache-aided UDN.

2.2 Caching policy

The caching policy, which is implemented only at the sBSs, improves the delay performance and offloads the traffic in the backhaul link. Suppose that the system is time slotted and that slot t is normalized to an integer unit within a short time interval $(t, t+1]$, where $t \in \{0, 1, 2, \ldots\}$. The sleeping strategy is operated over a long time period.

In each time slot t, the video and file contents requested by users must be transmitted through the system. The transmission service for file contents is provided by the mBSs. Each mBS makes all file contents available and is connected to the data center via a fiber backhaul link [16]. Meanwhile, each sBS is equipped with a local storage that caches all video contents. To reduce the delays in file transmission and offloading the mBS traffic load and given the limited cache space, we assume that each sBS can cache a subset of the file contents [16].

The set of file contents requested by the user is denoted by $\mathcal{F}=\{f_1, \ldots, f_{C_N}\}$, and the total number of file contents is C_N. The size of each file is assumed as L_F [10, 16]. The file popularities are modeled by a Zipf distribution with the parameter σ [16, 18]. The request probability of the ith most popular file content is then given by the following:

$$p_i = i^{-\sigma} \left(\sum_{i=1}^{C_N} i^{-\sigma} \right)^{-1}, \tag{1}$$

where $\sigma \geqslant 0$ reflects the popularity distribution of the contents. A high σ means that most of the users request a few very popular files.

We employ the most popular caching strategy, which assumes that sBS k can cache at most $C_k < C_N$ files (i.e., the cache capacity is C_k). The files cached at sBS are denoted as $\mathcal{C}_k=\{f_{C_1}, \ldots, f_{C_k}\}$. The caching probability of a user requesting a file content cached within sBS k is given by [16, 18]

$$p_k^C = \sum_{i=1}^{C_k} p_i. \tag{2}$$

2.3 Transmission model

In each time slot t, suppose that the number of requests for new video contents that randomly arrive at the system is an independent identically distributed (iid) PPP with arrival rate λ_V. The average size of each video is assumed as L_V [8]. Similarly, suppose that the number of new file requests arriving at the system is an iid PPP with arrival rate λ_F. As congestion of the transmission links and backhaul links in the mBS will delay the file-content transmission, users should receive the file contents from the sBS whenever possible. However, the probability p_k^F of the file content transmitted by the nearest sBS k depends on the state of sBS k and the cache capacity C_k.

Suppose that when no video request is being served, the file content can be transmitted by the nearest sBS k, which also caches the file content [16]; otherwise, the file content is delivered by the nearest mBS.

It is worth noting that when a video request is being served, the file content cannot be transmitted by an sBS. Meanwhile, when an sBS transmits its file contents to a user, the newly arrived video request is transmitted simultaneously, and the video and file contents share the same BS resource.

2.4 System energy consumption model

The BS consumes a fixed amount of energy, along with the load-dependent transmission energy and the storage energy [13, 15]. The fixed energy is the energy demand of the basic circuit, which is related to the type of BS, duration of the period, and the temperature. The load-dependent energy is proportional to the amount of data transmitted in the link, and the storage energy is consumed by the caching policy. The cache-aided UDN significantly reduces the backhaul energy consumption, but also consumes storage energy.

The power consumption (i.e., energy consumption) of the backhaul link from the data center to mBS j in slot t is expressed as [13]

$$P_j^M(t) = P_{m0} + \bar{\xi}_j \left(P_{mb} + \Delta p_m P_{mt}\right). \tag{3}$$

Meanwhile, the power (energy) consumption of cache-aided sBS k in slot t under a certain state $\boldsymbol{S}$ of the sBS is given as

$$P_k^S(t) = (1 - s_k) P_S + s_k \left(P_{s0} + \bar{\rho}_k \Delta p_s P_{st} + C_k L_F \omega_{cs}\right), \tag{4}$$

where s_k presents the state of sBS k, and P_S denotes the average power consumption of a sleeping sBS. P_{s0} ($P_{s0} \geqslant P_S$) and P_{m0} are the constant power consumptions of sBS and mBS in the active state, respectively, with corresponding power amplifying factors of Δp_s and Δp_m ($\Delta p_m \geqslant \Delta p_s$). P_{st} and P_{mt} ($P_{mt} \geqslant P_{st}$) are the transmission powers of sBS and mBS, respectively. The average traffic intensities in mBS j and sBS k are $\bar{\xi}_j$ and $\bar{\rho}_k$ respectively [8], and P_{mb} is the power consumption of the backhaul link to the mBS.

To compute the caching energy consumption, we separate the energy consumptions of caching the video and file contents. The energy consumption of video-caching is constant and denoted by sBS P_{s0}, whereas that of file-caching is given by $C_k L_F \omega_{cs}$ [16], where ω_{cs} denotes the caching efficiency.

As t is an integer unit slot, and the sleeping strategy is operated at the beginning of t, the energy consumed by the system in slot t for a certain state of sBS $\boldsymbol{S}$ is given as

$$P(t) = \sum_{j=1}^{N_m} P_j^M(t) + \sum_{k=1}^{N_s} P_k^S(t). \tag{5}$$

Obviously, increasing the number of sleeping sBSs reduces the system's energy consumption. Moreover, as the energy consumption of caching the file contents in an sBS is smaller than the transmission energy consumption of the mBS, the system energy consumption can be reduced by enlarging the cache capacity.

3 Queue model and mean delay

Suppose that each sBS and mBS have an unlimited capacity for queueing the users waiting for services. As the total request arrivals for video and file contents are PPPs, the requests at each active sBS and mBS in slot t are also PPPs with different arrival rates [19]. Assuming that the service rate (defined as the number of files or videos served per unit time) of each BS follows a general distribution, the transmission rate of sBS for a given state of sBS $\boldsymbol{S}$ can be obtained using the Shannon capacity formula. Consequently, we model each active sBS and mBS as a queueing system on a shared M/G/1 processor [20].

The mean delay in delivering the video and file contents is then derived from the average queue length and network stability condition in a queueing model based on Little's theorem (as described in the next subsection).

3.1 Queue model of sBS

Let $A_k(t)$ be the average arrival rate of requests for sBS k in slot t; i.e., the number of new video requests arriving at sBS k and the number of user-received file transmission services at sBS k in slot t:

$$A_k(t) = s_k \left(\frac{\lambda_V N_s}{\lambda_s |\mathcal{B}_{\text{on}}|} + p_k^F \frac{\lambda_F N_s}{\lambda_s |\mathcal{B}_{\text{on}}|} \right), \tag{6}$$

where s_k represents the state of sBS k, and p_k^F is the probability of the file content transmitted by the nearest sBS k.

In such a Poisson traffic model, the discrete transmission requests arrive at sBSs as a PPP. The traffic intensity (or system utilization [8, 20]) is defined as the probability that sBS k is busy in an active state in slot t,

$$\rho_k(t) = \frac{A_k(t)}{\mu_k(t)}, \tag{7}$$

where $\mu_k(t)$ is the service rate of sBS k in slot t. By definition of traffic intensity, the probability that sBS k is idle when no video contents are waiting for service at sBS k is $1 - \lambda_V N_s L_V / (\lambda_s |\mathcal{B}_{\text{on}}| R_k(t))$.

Furthermore, the probability that sBS k can service a file transmission is

$$p_k^F = s_k \left(1 - \frac{\lambda_V N_s}{\lambda_s |\mathcal{B}_{\text{on}}|} \frac{L_V}{R_k(t)} \right) p_k^C. \tag{8}$$

The service rate $\mu_k(t)$ of sBS k in slot t is then given by the following:

$$\mu_k(t) = s_k \left(\frac{\lambda_V N_s}{\lambda_s |\mathcal{B}_{\text{on}}|} + \left(1 - \frac{\lambda_V N_s}{\lambda_s |\mathcal{B}_{\text{on}}|} \frac{L_V}{R_k(t)} \right) \frac{R_k(t)}{L_F} \right). \tag{9}$$

We now define the queue length $Q_k(t)$, the average number of video contents of the users waiting for transmission in the buffer of sBS k in slot t [21], as

$$Q_k(t) = \max\{0, Q_k(t-1) - R_k(t)\} + A_k(t). \tag{10}$$

The time-averaged arrival rate, transmission rate, queue length, and traffic intensity at sBS k are respectively defined as follows: [7]

$$\bar{A}_k = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathrm{E}\{A_k(t)\}, \tag{11}$$

$$\bar{R}_k = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathrm{E}\{R_k(t)\}, \tag{12}$$

$$\bar{Q}_k = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathrm{E}\{Q_k(t)\}, \tag{13}$$

$$\bar{\rho}_k = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathrm{E}\{\rho_k(t)\}. \tag{14}$$

3.2 Queue model of mBS

The number of new file requests from users arriving at mBS j in slot t, i.e., the arrival rate of mBS j, is given as follows:

$$M_j(t) = \frac{\lambda_F}{\lambda_m}\left(1 - \sum_{k=1}^{N_s} p_k^F \frac{N_s}{\lambda_s |\mathcal{B}_{\text{on}}|}\right). \tag{15}$$

Denote the average transmission rate of the mBS j files provided to users in slot t by $r_j(t)$. The service rate of mBS j in slot t is given by $\chi_j(t) = r_j(t)/L_F$.

The traffic intensity of mBS j in slot t is then given as

$$\xi_j(t) = \frac{M_j(t)}{\chi_j(t)}. \tag{16}$$

The queue length of mBS j in slot t can be expressed as

$$G_j(t) = \max\{0, G_j(t-1) - r_j(t)\} + M_j(t). \tag{17}$$

The time-averaged arrival rate, transmission rate, queue length, and traffic intensity at mBS j are respectively given by the following equations:

$$\bar{M}_j = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathrm{E}\{M_j(t)\}, \tag{18}$$

$$\bar{r}_j = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathrm{E}\{r_j(t)\}, \tag{19}$$

$$\bar{G}_j = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathrm{E}\{G_j(t)\}, \tag{20}$$

$$\bar{\xi}_j = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{\mathrm{T}-1} \mathrm{E}\{\xi_j(t)\}. \tag{21}$$

3.3 Network stability

A discrete queueing process for sBS k is mean rate stable if [21]

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathrm{E}\{Q_k(t)\} < \infty. \tag{22}$$

Similarly, the queueing process at mBS j is mean rate stable if

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathrm{E}\{G_j(t)\} < \infty. \tag{23}$$

A network is stable if all queues are mean rate stable, which implies that the time-averaged arrival process is less than or equal to the service process, i.e., all queues are finitely long [7].

3.4 Mean delay

In this paper, delay is defined as the time difference between the arrival of a user's transmission request and the departure of the user's service.

The delay in a video content request comprises the transmission delay and the queueing delay. The average transmission delay of the video content transmitted by sBS k in slot t is given by

$$D_T^V(t) = \frac{L_V}{\bar{R}_k}, \tag{24}$$

where $\bar{R}_k$ is the time-averaged transmission rate of serving sBS k.

According to Little's theorem [20], the average queueing delay of the video content transmitted by sBS k in slot t is given as

$$D_Q^V(t) = \frac{\bar{Q}_k}{\bar{A}_k}. \tag{25}$$

Thus, the mean delay of the video content transmitted by sBS k in slot t can be expressed as follows:

$$D^V(t) = D_T^V(t) + D_Q^V(t). \tag{26}$$

File contents can be transmitted by the nearest sBS k without a queueing delay, or by the nearest mBS j with both transmission and queueing delays. In addition, the backhaul link from a given mBS to the data center is a dedicated backhaul link with low delay. This delay can be neglected [22].

Therefore, the mean delay of delivering the file content in slot t is given as [23]

$$D^F(t) = p_k^F \frac{L_F}{\bar{R}_k} + \left(1 - p_k^F\right)\left(\frac{L_F}{\bar{r}_j} + \frac{\bar{G}_j}{\bar{M}_j}\right). \tag{27}$$

The time-averaged mean delay of delivering the video and file contents is then expressed as

$$\bar{D}^V = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathrm{E}\left\{D^V(t)\right\}, \tag{28}$$

$$\bar{D}^F = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathrm{E}\left\{D^F(t)\right\}. \tag{29}$$

As evidenced in the above analysis (Eqs. (6) and (15)), enlarging the number of sleeping sBSs in the system increases the number of user requests at each BS, and hence lengthens the queue [24]. Additionally, at the given traffic arrival rate, the delays in the video and file contents are proportional to the corresponding queue lengths (Eqs. (26) and (27)). Thus, the queue length can be depicted as a delay and can be controlled within a tolerable limit by adjusting the sleeping mechanism [7].

If the file content is cached at an sBS, it can be transmitted directly from the BS to a user with high p_k^F, so the file-content delay decreases with increasing cache capacity for the file contents. Nevertheless, transmitting a file content delays the video transmission owing to the increased p_k^F and $\bar{Q}_k$, meaning that increasing the cache capacity for the file content increases the delay in delivering the video content.

4 Delay-constrained sleeping mechanism for the energy saving problem

This section aims to optimize the sleeping set of the sleeping mechanism in the cache-aided UDN, thus reducing the energy consumption of the network while ensuring good delay performance.

4.1 Problem formulation

According to the queueing model and Little's theorem, the queueing delay depends on the queue length, which is related to the network stability. In a stable network, all queues are of finite length. Therefore, we depict the delay by the queue length and guarantee a finite delay by the network stability condition.

The energy saving problem is formulated as a minimization problem of the time-averaged energy consumption of the system subject to a network stability criterion:

$$\mathrm{P1} \qquad \arg\min_{\boldsymbol{S}^*} \bar{P} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathrm{E}\left\{P(t)\right\}, \tag{30}$$

$$\text{s.t.} \quad \mathrm{C1}: \quad \bar{Q} + \bar{G} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \left(\sum_{k=1}^{N_s} Q_k(t) + \sum_{j=1}^{N_m} G_j(t) \right) < \infty, \tag{31}$$

where C1 is the network stability condition that guarantees a suitable delay.

The objective function P1 is the long-term time average of the expected energy consumption under the network stability constraint. Owing to the binary-mode selection of the sBS state, the objective function P1 cannot be solved directly.

Instead, a mutual restraint relationship between the energy consumption and delay is found using the Lyapunov optimization theory, and the original problem P1 is transformed into a joint optimization problem that simultaneously solves the energy consumption and delay.

4.2 Problem transformation

Lyapunov optimization is a classic approach that has been widely applied in recent energy efficiency-delay tradeoff problems [7,21,25,26]. It is particularly efficient at optimizing the time-averaged objective function under additional time-averaged constraints [27], as the original stochastic optimization problem can be transformed into an instantaneous static optimization problem. Thus, the stochastic optimization problem P1 can be directly solved by a classical drift-plus-penalty algorithm developed by the Lyapunov optimization technique.

Let $\Theta(t)=\{Q(t), G(t)\}$. The Lyapunov function is defined as

$$L(\Theta(t)) = \frac{1}{2}\sum_{k=1}^{N_s} Q_k(t)^2 + \frac{1}{2}\sum_{j=1}^{N_m} G_j(t)^2, \tag{32}$$

where $G(t)=\{G_1(t),\ldots,G_j(t),\ldots,G_{N_m}(t)\}$ and $Q(t)=\{Q_1(t),\ldots,Q_k(t),\ldots,Q_{N_s}(t)\}$.

The one-slot conditional Lyapunov drift is then given by the following equation:

$$\Delta(\Theta(t)) = \mathrm{E}\{L(\Theta(t+1)) - L(\Theta(t)) \,|\Theta(t)\}. \tag{33}$$

Theorem 1. According to the Lyapunov optimization technique, under any state of sBS $\boldsymbol{S}$, all possible values of $\Delta(\Theta(t))$ and all possible weighting factors $V>0$, P1 can be solved by minimizing the upper bound of the drift-plus-penalty expression [7]:

$$\begin{aligned}\Delta(\Theta(t)) + V\mathrm{E}\{P(t)\,|\Theta(t)\} \leqslant B + V\mathrm{E}\{P(t)\,|\Theta(t)\} + \sum_{k=1}^{N_s} Q_k(t)\left(\mathrm{E}(A_k(t)\,|\Theta(t)) \right.\\ \left. -\mathrm{E}(R_k(t)\,|\Theta(t))\right) + \sum_{j=1}^{N_m} G_j(t)\left(\mathrm{E}(M_j(t)\,|\Theta(t)) - \mathrm{E}(r_j(t)\,|\Theta(t))\right),\end{aligned} \tag{34}$$

where B is a positive constant, and for all time slots t we have

$$B \geqslant \frac{1}{2}\sum_{k=1}^{N_s}\mathrm{E}\left\{A_k(t)^2 + R_k(t)^2\,|\Theta(t)\right\} + \frac{1}{2}\sum_{j=1}^{N_m}\mathrm{E}\left\{M_j(t)^2 + r_j(t)^2\,|\Theta(t)\right\}. \tag{35}$$

Proof. A similar proof is given in [7,21].

By Theorem 1, minimizing the upper bound of the drift-plus-penalty expression also minimizes the Lyapunov drift $\Delta(\Theta(t))$, ensuring that all queues are stable. In other words, to satisfy the network stability condition C1, we need only to minimize the upper bound of the drift-plus-penalty expression, denoted by $F_S(t)$. Therefore, we transform P1 as follows:

$$\begin{aligned}\mathrm{P2}\quad \arg\min_{\boldsymbol{S}^*}\quad F_S(t) = \Bigg\{ & V\mathrm{E}\{P(t)\,|\Theta(t)\} + \sum_{k=1}^{N_s} Q_k(t)\left(\mathrm{E}(A_k(t)\,|\Theta(t)) - \mathrm{E}(R_k(t)\,|\Theta(t))\right) \\ & + \sum_{\mathrm{j}=1}^{N_m} G_j(t)\left(\mathrm{E}(M_j(t)\,|\Theta(t)) - \mathrm{E}(r_j(t)\,|\Theta(t))\right)\Bigg\},\end{aligned} \tag{36}$$

where the weighting factor V can be set to any positive value. The weighting factor indicates the relative importance of the average energy consumption over the mean delay [8].

Note that the optimization problem P2 is not equivalent to the original problem P1, as will be shown in Theorem 2. However, by setting a very large weighting factor V, problems P1 and P2 become equivalent. In addition, to reveal the relationship between the energy consumption and delay, we bound the sum of the time-averaged queue lengths by Theorem 3.

Theorem 2. Suppose that the original problem is feasible. For some arbitrary $V > 0$, the time-averaged energy consumption in P2 is bounded by

$$\bar{P}_{P1}^{\text{opt}} \leqslant \bar{P}_{P2}^{\text{opt}} \leqslant \bar{P}_{P1}^{\text{opt}} + \frac{B}{V}, \tag{37}$$

where $\bar{P}_{P1}^{\text{opt}}$ and $\bar{P}_{P2}^{\text{opt}}$ are the time-averaged energy consumptions in the optimal solutions of P1 and P2, respectively.

Proof. As in [7, 21], suppose that $R_k(t)$, $r_j(t)$ and $A_k(t)$, $M_j(t)$ are derived for any arbitrary state of sBS $\boldsymbol{S}$, and satisfy the following conditions for any $\varepsilon > 0$,

$$\mathrm{E}\left(A_k\left(t\right)|\Theta\left(t\right)\right) = \lambda_k^S, \quad \mathrm{E}\left(M_j\left(t\right)|\Theta\left(t\right)\right) = \lambda_j^M, \tag{38}$$

$$\mathrm{E}\left(R_k\left(t\right)|\Theta\left(t\right)\right) = \mathrm{E}\left(R_k\left(t\right)\right) \geqslant \lambda_k^S + \varepsilon, \tag{39}$$

$$\mathrm{E}\left(r_j\left(t\right)|\Theta\left(t\right)\right) = \mathrm{E}\left(r_j\left(t\right)\right) \geqslant \lambda_j^M + \varepsilon. \tag{40}$$

Assume that the time-averaged energy consumption in the optimal solution of P2 $\bar{P}_{P2}^{\text{opt}}$ is bounded by $\bar{P}_{\min} \leqslant \bar{P}_{P2}^{\text{opt}} \leqslant \bar{P}_{\max}$.

Substituting (38)–(40) into (36) and taking $\varepsilon \to 0$ yields

$$\Delta\left(\Theta\left(t\right)\right) + V\mathrm{E}\left\{\bar{P}_{P2}^{\text{opt}}\left(t\right)|\Theta\left(t\right)\right\} \leqslant B + V\bar{P}_{P1}^{\text{opt}} - \varepsilon\sum_{k=1}^{N_s} Q_k\left(t\right) - \varepsilon\sum_{j=1}^{N_m} G_j\left(t\right). \tag{41}$$

Iterating the expectation and telescoping the sums over $t \in \{1, \ldots, T\}$, we have

$$\mathrm{E}\{L\left(\Theta\left(T+1\right)\right)\} - \mathrm{E}\left\{L\left(\Theta\left(T\right)\right)\right\} + V\sum_{t=1}^{T}\mathrm{E}\left\{\bar{P}_{P2}^{\text{opt}}\left(t\right)|\Theta\left(t\right)\right\} \leqslant T\left(B + V\bar{P}_{P1}^{\text{opt}}\right) - \varepsilon\sum_{t=1}^{T}\left(\sum_{k=1}^{N_s} Q_k\left(t\right) + \sum_{j=1}^{N_m} G_j\left(t\right)\right). \tag{42}$$

Dividing (42) by VT and letting $T \to \infty$, we obtain (37).

Theorem 2 demonstrates that as V increases, the time-averaged energy consumption in the optimal solution of P2 $\bar{P}_{P2}^{\text{opt}}$ decreases at the speed $O\left(1/V\right)$. Specially, according to (37), $\bar{P}_{P2}^{\text{opt}}$ is arbitrarily close to the optimal solution of the original problem when V is sufficiently large. Hence, after setting a very large weighting factor V, the original problem P1 becomes equivalent to the optimization problem P2.

Moreover, as sleeping BSs increase the delay, there exists a tradeoff between energy consumption and delay. To understand the relationship between the energy consumption and mean delay, we bounded the summed average queue lengths using Theorem 3, obtained by dividing (42) by εT and letting $T \to \infty$.

Theorem 3. Suppose that the original problem P1 is feasible. For arbitrary $V > 0$, the optimal solution of P2 guarantees that all queues in the system are stable, i.e., Eq. (31) holds. The sum of the time-averaged queue lengths is bounded by

$$\bar{Q} + \bar{G} = \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\left(\sum_{k=1}^{N_s} Q_k\left(t\right) + \sum_{j=1}^{N_m} G_j\left(t\right)\right) \leqslant \frac{1}{\varepsilon}\left(B + V\bar{P}_{\max}\right). \tag{43}$$

Theorem 3 guarantees the finiteness of all queues in the optimal solution of P2, but the sum of the time-averaged queue lengths increases at the speed $O\left(V\right)$ with increasing V, meaning that a large V lengthens the delay. Theorems 2 and 3 imply that V balances the energy consumption and delay. More specifically, the energy consumption becomes arbitrarily close to the optimal solution of P1 when V is set large enough to ensure an arbitrarily small B/V. On the contrary, a small V is required for minimizing the mean delay; that is, for reducing the queue length as far as possible .

4.3 Solution to optimization problem P2

Problem P2 must find the optimal sleeping set that minimizes the objective function $F_S(t)$ in each slot t. As problem P2 is a challenging combinatorial optimization problem, its optimal solution can be found by exhaustive searching over the $O\left(2^{N_s}\right)$ possible cases. Here, the set of sleeping sBSs that saves energy while maintaining delay performance was decided by a low-complexity, delay-constrained sleeping strategy. The proposed strategy is presented in Algorithm 1.

We now introduce $\Gamma_{\text{on}}(k)$ and $\Gamma_{\text{off}}(k)$, the turn-on and turn-off benefits of $F_S(t)$, respectively, for sBS k, and denote by $F(\mathcal{B}_{\text{on}})$ the value of $F_S(t)$ given the active set sBS $\mathcal{B}_{\text{on}}$. As a sleeping BS consumes more energy when awakened, we consider the energy consumption E_S of transiting from the active to the sleeping state [28].

Algorithm 1 first calculates the average transmission rate, average queue length and average energy consumption at each BS, based on the current set $\mathcal{B}_{\text{on}}$ of active sBSs. These calculations are iterated over the T time slots (Steps 2–10).

For the given set $\mathcal{B}_{\text{on}}$ of active sBSs, it then checks the network stability condition according to the traffic intensity $\Psi = (\rho_1, \ldots, \rho_{N_s}, \xi_1, \ldots, \xi_{N_m})$.

If $\Psi > 1$ (i.e., the service rate is smaller than the arrival rate), there is an unstable queue in the system. To ensure a stable network, we awaken some sleeping sBSs depending on the value of $F_S(t)$ (Steps 11–16). In constructing $\mathcal{B}_{\text{on}}$, we choose the sBS k that maximizes the turn-on benefit $\Gamma_{\text{on}}(k)$. In particular, we turn on the sBS k that minimizes the objective function $F_S(t)$ and add it to the previous set $\mathcal{B}_{\text{on}}$ of active sBSs. These processes repeat until the network stability condition is satisfied.

If $0 < \Psi < 1$, (i.e., the network stability condition is satisfied), we calculate the turn-off benefit $\Gamma_{\text{off}}(k)$ for each active sBS k in $\mathcal{B}_{\text{on}}$. If $\Gamma_{\text{off}}(k) > 0$ (i.e., if turning off sBS k benefits the system), the selected sBS k is added to the set $\mathcal{B}_{\text{off}}$ of sleeping sBSs; otherwise, the algorithm stops (Steps 17–20).

The detailed sleeping mechanism is presented in Algorithm 1. The complexity of this algorithm is $O((T+1)N_s + TN_m)$.

Algorithm 1 A delay-constrained sleeping scheme

01: **Initialize:** $Q(0)=0, G(0)=0, \rho(0)=0, \xi(0)=0, \mathcal{B}_{\text{on}}, \mathcal{B}_{\text{off}}$.
02: **Repeat**
03: Calculate $R_k(t)$, $r_j(t)$, $\rho_k(t)$, $\xi_j(t)$ and p_k^F.
04: Compute $A_k(t)$ and $M_j(t)$ according to (6) and (15).
05: Update $Q_k(t)$ and $G_j(t)$ according to (10) and (17).
06: Calculate $P(t)$ according to (5).
07: $t = t + 1$,
08: Update $\bar{\rho}_k$ and $\bar{\xi}_j$ according to (14) and (21).
09: Update $\bar{Q}_k$, $\bar{G}_j$ and $\bar{P}$, according to (13), (20) and (30).
10: **End Repeat** when $t = T$, T is total number of time slots.
11: **While** $\Psi > 1$,
12: **Repeat**
13: Calculate $\Gamma_{\text{on}}(k) = F(\mathcal{B}_{\text{on}} \cup \{k\}) - F(\mathcal{B}_{\text{on}}) + E_s$, $\forall k \in \mathcal{B}_{\text{off}}$.
14: **If** $k^* = \arg\min_{k \in \mathcal{B}_{\text{on}}} \Gamma_{\text{on}}(k)$, **Then** $\mathcal{B}_{\text{on}} \leftarrow \mathcal{B}_{\text{on}} \cup \{k^*\}$.
15: **End Repeat** when $0 < \Psi < 1$.
16: **End While**
17: **While** $0 < \Psi < 1$,
18: Calculate $\Gamma_{\text{off}}(k) = F(\mathcal{B}_{\text{on}}) - F(\mathcal{B}_{\text{on}} - \{k\}) - E_s$, $\forall k \in \mathcal{B}_{\text{on}}$.
19: **If** $\Gamma_{\text{off}}(k) > 0$, **Then** $\mathcal{B}_{\text{off}} \leftarrow \mathcal{B}_{\text{off}} \cup \{k\}$.
20: **End While**

5 Numerical and simulation results

This section presents the numerical results of the simulation study. The system covers 250000π m^2, and the sBS and mBS-tiers operate on different frequencies. We assume that all active sBSs have the same

Table 1 System parameter

Parameter	Value	Parameter	Value
λ_m	5×10^{-6}	λ_s	2.5×10^{-5}
λ_V	0.5 s^{-1}	λ_F	1 s^{-1}
Δp_m	10	Δp_s	8
P_{s0}	4.8 W	P_{m0}	10 W
P_S	2.4 W	P_{mb}	8 W
P_{mt}	46 dBm	P_{st}	30 dBm
L_V	10 MB	L_F	5 MB
W_m	10 MHz	W_s	10 MHz
E_S	1.5 W	ω_{cs}	2×10^{-9} J/byte

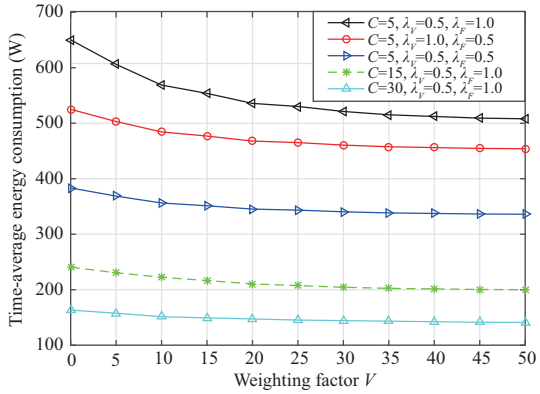

Figure 2 (Color online) Time-averaged energy consumption vs. weighting factor V, calculated by Algorithm 1.

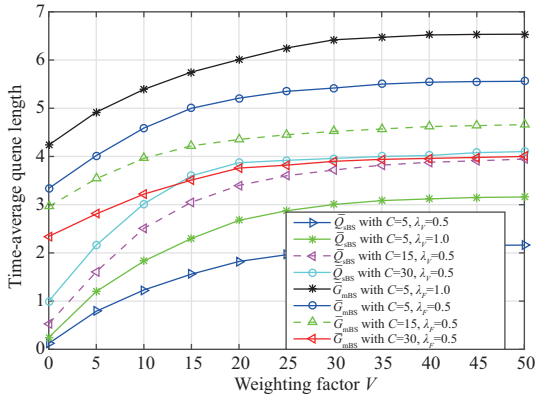

Figure 3 (Color online) Time-averaged queue length vs. weighting factor V, obtained in the simulation study.

cache capacity C with parameter $\sigma = 0.5$. The system parameters are consistent with those in [8,16,29], and are listed in Table 1.

Figure 2 plots the numerically calculated time-averaged energy consumption vs. the weighting factor V under the optimal sBS state obtained by Algorithm 1. Results are plotted for five combinations of C, λ_m, and λ_V. As V increased, the time-averaged energy consumption decreased at the speed $O(1/V)$, verifying Theorem 2. In particular, when V was sufficiently large (e.g., $V > 40$), the time-averaged energy consumption converged to the optimal solution of problem P2. In addition, increasing C obviously decreased the energy consumption, because increasing the C reduces the traffic load of the mBS; moreover, caching the file contents of an sBS consumes less energy than transmitting the file contents from an mBS. Caching can induce the sleeping state in some sBSs. Therefore, caching can decrease the intensity of active BSs in the UDN [30,31].

Decreasing the traffic arrival rate λ_V, λ_F also reduced the time-averaged energy consumption (see Figure 2). A lower traffic-arrival rate lightens the traffic load at each BS; accordingly, the time-averaged energy consumption was smaller for $\lambda_V = 0.5$, $\lambda_F = 0.5$ than for $\lambda_V = 1.0$, $\lambda_F = 0.5$ and $\lambda_V = 0.5$, $\lambda_F = 1.0$. In particular, as transmission from an mBS consumes more average energy than transmission from an sBS, the time-averaged energy consumption was larger for $\lambda_V = 0.5$, $\lambda_F = 1.0$ than for $\lambda_V = 1.0$, $\lambda_F = 0.5$.

Figure 3 plots the simulated time-averaged queue lengths at sBSs and mBSs as functions of V for different C values. As V increased, the time-averaged queue lengths of the sBS and mBS ($\bar{Q}_{\mathrm{sBS}}$ and $\bar{G}_{\mathrm{mBS}}$, respectively) increased at the speed $O(V)$, verifying Theorem 3. In particular, when V was sufficiently large (e.g., $V > 35$), the time-averaged queue lengths converged to the optimal solution of problem P2.

As also confirmed in Figure 3, increasing C increased the time-averaged queue length of the sBS but shortened the queue length of mBS. This occurs because as C increases, a larger number of file contents

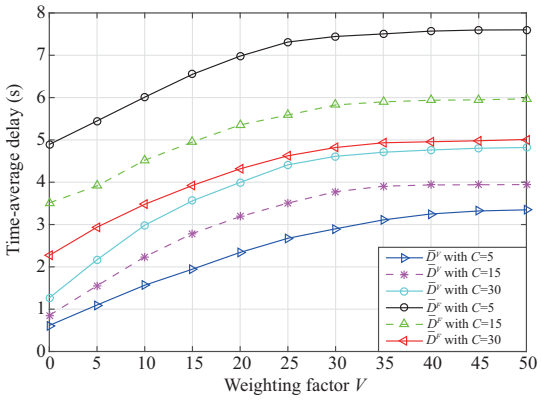

Figure 4 (Color online) Time-averaged delays in file and video contents vs. weighting factor V, obtained in the simulation.

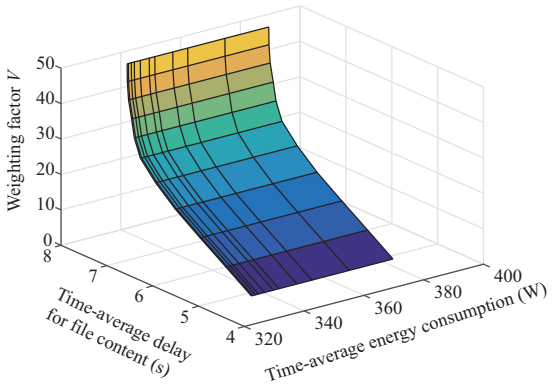

Figure 5 (Color online) Weighting factor V vs. the time-averaged energy consumption and the time-averaged delay of file contents at $C = 5$.

are accepted for service at each active sBS, creating a longer queue of video contents waiting for service at the sBSs.

We further observed that the time-average queue length of sBS was larger for $C = 5$, $\lambda_V = 1.0$ than for $C = 5$, $\lambda_V = 0.5$. This result reflects Little's theorem, namely, that the average queue length is proportional to the average arrival rate of the file contents. The same trend appeared for the time-averaged queue length of mBS (Figure 3).

Figure 4 plots the simulated time-averaged delays in file and video contents as functions of V for different C values. Obviously, the delays in the video and file contents were proportional to the average queue lengths at the sBSs and mBSs, respectively. Additionally, the delay curves in Figure 4 exhibit almost the same trend as the queue-length curves in Figure 3.

Figure 5 plots the weighting factor V as a function of the time-averaged energy consumption and the time-averaged delay of the file contents for $C = 5$. Obviously, increasing the V decreased the energy consumption and lengthened the delay in transmitting the file content. Thus, the relationship between the energy consumption and file-content delay deviates from a monotonic curve, meaning that sacrificing the delay does not guarantee an energy-saving benefit.

Figure 6 compares the time-averaged energy consumption vs. weighting factor curves in different sleeping schemes and caching capacities. As a benchmark, the exhaustive-search results of the original problem P1 are also presented. The time-averaged energy consumptions obtained by our proposed Algorithm 1 closely matched those found by exhaustive searching. Comparing the schemes with and without the sleeping strategy, we observed that the sleeping strategy improved the energy conservation performance. Combining the sleeping strategy with the caching scheme further reduced the energy consumption by a considerable amount.

Note also that the time-averaged energy consumption for $C = 5$ with no sleeping strategy was larger than that for $C = 0$ with the sleeping strategy because the reduction of the average energy consumption by the sleeping strategy is larger than that of caching file content for sBS when $C = 5$.

Figure 7 compares the simulated time-averaged delays in transmitting the file and video contents as functions of weighting factor in different sleeping schemes and for different caching capacities. This figure clarifies an increasing trend in the mean delays of video and file contents as the weighting factor V increased. Meanwhile, when V was sufficiently large (e.g., $V > 35$), the time-averaged delay converged to the optimal solution of problem P2, demonstrating that the condition C1 was satisfied.

The results of our proposed Algorithm 1 were very similar to those of exhaustive searching. Benefitting from the caching strategy, the cache-aided UDN improved the mean delay performance of delivering the file content (compare the results of $C = 5$ and $C = 0$ obtained by Algorithm 1). On the contrary, the video-content delivery might be compromised because the file and video contents compete for the same sBS resources.

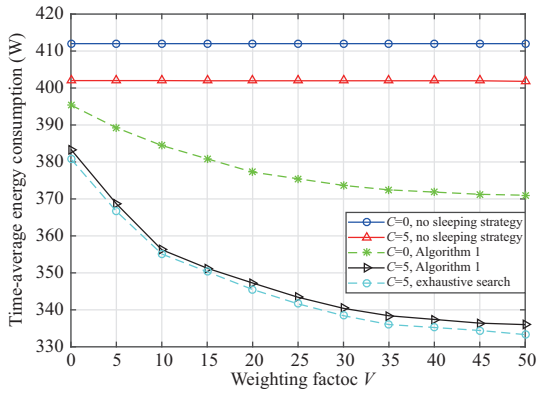

Figure 6 (Color online) Time-averaged energy consumption vs. weighting factor for different sleeping schemes and caching capacities.

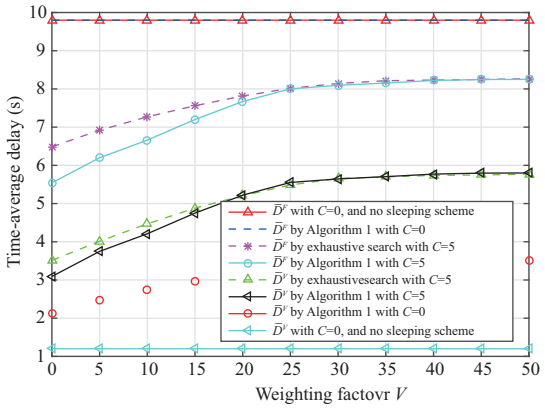

Figure 7 (Color online) Time-averaged delay vs. weighting factor for different sleeping schemes and caching capacities.

In addition, the sleeping strategy clearly extended the delay of video content, meaning that the delay performance of video contents was improved by removing the sleeping strategy. On the other hand, as the mBS is responsible for providing file services, the weighting factor did not influence the file-content delay when $C = 0$, regardless of whether the sleeping mechanism was adopted or omitted.

6 Conclusion

This paper investigated an energy saving problem for a cache-aided UDN with a sleeping strategy and subjected to a delay constraint. The system energy consumption, mean delay in file delivery, and video transmission to users were analyzed using a mathematical model. The energy saving problem was reformulated as an energy minimization problem constrained by a network stability condition. Using the Lyapunov optimization technique, the objective function was transformed into the joint optimization problem of energy consumption and delay with a weighting factor. The joint optimization problem was then solved by a delay-constrained sleeping strategy. Our proposed strategy was verified by the numerical results of a simulation study. Our proposed algorithm delivered good performance, and achieved a suitable tradeoff between the energy consumption and delay through the adjustable weighting factor.

Acknowledgements This work was partially supported by National Major Project (Grant No. 2017ZX03001002-004), National Natural Science Foundation Project of China (Grant No. 61521061), and 333 Program of Jiangsu (Grant No. BRA2017366).

References

1 You X H, Pan Z W, Gao X Q, et al. The 5G mobile communication: the development trends and its emerging key techniques (in Chinese). Sci Sin Inform, 2014, 44: 551–563
2 Wu J, Zhang Y, Zukerman M, et al. Energy-efficient base-stations sleep-mode techniques in green cellular networks: a survey. IEEE Commun Surv Tut, 2015, 17: 803–826
3 Liu C, Natarajan B, Xia H. Small cell base station sleep strategies for energy efficiency. IEEE Trans Veh Technol, 2016, 65: 1652–1661
4 Liu B, Zhao M, Zhou W, et al. Flow-level-delay constrainted small cell sleeping with macro base station cooperation for energy saving in HetNet. In: Proceedings of IEEE Vehicular Technology Conference (VTC-Fall), Boston, 2015. 1–5
5 Gamboa S, Pelov A, Maille P, et al. Reducing the energy footprint of cellular networks with delay-tolerant users. IEEE Syst J, 2017, 11: 729–739
6 Son K, Kim H, Yi Y, et al. Base station operation and user association mechanisms for energy-delay tradeoffs in green cellular networks. IEEE J Sel Areas Commun, 2011, 29: 1525–1536
7 Shi Q, Zhao L, Zhang Y, et al. Energy-efficiency versus delay tradeoff in wireless networks virtualization. IEEE Trans Veh Technol, 2018, 67: 837–841

8 Li P, Jiang H L, Pan Z W, et al. Energy-delay tradeoff in ultra-dense networks considering BS sleeping and cell association. IEEE Trans Veh Technol, 2018, 67: 734–751
9 Han T, Ansari N. Network utility aware traffic load balancing in backhaul-constrained cache-enabled small cell networks with hybrid power supplies. IEEE Trans Mobile Comput, 2017, 16: 2819–2832
10 Chen Z, Lee J, Quek T Q S, et al. Cooperative caching and transmission design in cluster-centric small cell networks. IEEE Trans Wirel Commun, 2017, 16: 3401–3415
11 Xu J W, Ota K, Dong M X. Saving energy on the edge: in-memory caching for multi-tier heterogeneous networks. IEEE Commun Mag, 2018, 56: 102–107
12 Melike E. Content caching in small cells with optimized uplink and caching power. In: Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, 2015. 2173–2178
13 Poularakis K, Iosifidis G, Tassiulas L. Joint caching and base station activation for green heterogeneous cellular networks. In: Proceedings of IEEE International Conference on Communications (ICC), London, 2015. 3364–3369
14 Xie R C, Li Z S, Huang T, et al. Energy-efficient joint content caching and small base station activation mechanism design in heterogeneous cellular networks. China Commun, 2017, 14: 70–83
15 Xu D, Jin H, Zhao C L, et al. Joint caching and sleep-active scheduling for energy-harvesting based small cells. In: Proceedings of IEEE International Conference on Wireless Communications and Signal Processing, Nanjing, 2017. 1–6
16 Pappas N, Chen Z, Dimitriou I. Throughput and delay analysis of wireless caching helper systems with random availability. IEEE Access, 2018, 6: 9667–9678
17 Doan K N, van Nguyen T, Quek T Q S, et al. Content-aware proactive caching for backhaul offloading in cellular network. IEEE Trans Wireless Commun, 2018, 17: 3128–3140
18 Gao S, Li P, Pan Z W, et al. Machine learning based small cell cache strategy for ultra dense networks. In: Proceedings of IEEE International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, 2017. 1–5
19 Haenggi M, Andrews J G, Baccelli F, et al. Stochastic geometry and random graphs for the analysis and design of wireless networks. IEEE J Sel Areas Commun, 2009, 27: 1029–1046
20 Takagi H. Queueing Analysis: A Foundation of Performance Evaluation, Volume I: Vacation and Priority Systems. 1st ed. Amsterdam: Elsevier, 1991. 30–55
21 Yang J, Yang Q H, Kwak K S, et al. Power-delay tradeoff in wireless powered communication networks. IEEE Trans Veh Technol, 2017, 66: 3280–3292
22 Zhang G Z, Quek T, Huang A, et al. Backhaul-aware base station association in two-tier heterogeneous cellular networks. In: Proceedings of IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Stockholm, 2015. 390–394
23 Li P, Shen Y, Sahito F, et al. BS sleeping strategy for energy-delay tradeoff in wireless-backhauling UDN. Sci China Inf Sci, 2019, 62: 042303
24 Zhu W X, Xu P P, Bui T O, et al. Energy-efficient cell-association bias adjustment algorithm for ultra-dense networks. Sci China Inf Sci, 2018, 61: 022306
25 Huang S, Liang B, Li J. Distributed interference and delay aware design for D2D communication in large wireless networks with adaptive interference estimation. IEEE Trans Wirel Commun, 2017, 16: 3924–3939
26 Mo Y, Peng M, Xiang H Y, et al. Resource allocation in cloud radio access networks with device-to-device communications. IEEE Access, 2017, 5: 1250–1262
27 Neely M J. Stochastic network optimization with application to communication and queueing systems. In: Synthesis Lectures on Communication Networks. San Rafael: Morgan and Claypool, 2010. 1–211
28 Wang C W, Mei W Y, Qin X Y, et al. Quantum entropy based tabu search algorithm for energy saving in SDWN. Sci China Inf Sci, 2017, 60: 040307
29 Jo H S, Xia P, Andrews J G. Open, closed, and shared access femtocells in the downlink. J Wirel Commun Netw, 2012, 2012: 363–378
30 Li K, Yang C, Chen Z, et al. Optimization and analysis of probabilistic caching in N -tier heterogeneous networks. IEEE Trans Wirel Commun, 2018, 17: 1283–1297
31 Cui Q M, Cui Z Y, Zheng W, et al. Energy-aware deployment of dense heterogeneous cellular networks with QoS constraints. Sci China Inf Sci, 2017, 60: 042303