EPS: Energy-efficient Pricing and Resource Scheduling in LTE-A Heterogeneous Networks

You-Chiun Wang and Kai-Chung Chien

Abstract—Many operators offer long term evolution–advanced (LTE-A) service for broadband wireless access, where they deploy diverse base stations to form a heterogeneous network (HetNet). It is critical to manage downlink resource to improve LTE-A performance. The paper studies the issues of energy saving and dynamic pricing in resource scheduling, and proposes an *Energy-efficient Pricing and resource Scheduling (EPS)* framework. It considers a HetNet scenario where picocells are densely deployed in each macrocell, and divides users into three classes for charge. EPS clusters picocells into groups, and selects a coordinator to arrange the service in each group, so as to share loads of picocells and save energy of base stations. Then, it adopts a two-layer scheduling strategy to allot resource to each flow based on its user class, channel quality, and packet delay. By using peak and off-peak rates, EPS adaptively adjusts the amount of money charged to each user to balance between operator profit and network utilization. Simulation results verify that EPS keeps high profit and throughput, and also saves more energy in LTE-A HetNets.

Index Terms—energy saving, heterogeneous network (HetNet), LTE-A system, pricing, resource scheduling.

1 Introduction

Mobile phones have become our main communication and computing devices today. Apart from voice calls, people also access the Internet via mobile phones, especially to use high data rate applications like multimedia streaming or video downloads [1]. This demand promotes the development of *long term evolution–advanced (LTE-A)*, which is the chief protocol for current mobile networks. LTE-A divides its downlink resource into units of *physical resource blocks (PRBs)*. Based on the channel quality of each user equipment (UE) to a PRB, the PRB can send different amount of data. PRBs are usually exclusive[1], which means that no two UEs can share the same PRBs. Since LTE-A does not specify how to allocate PRBs to UEs, various resource scheduling methods are developed to improve system performance [2].

The current trend is towards *heterogeneous network (HetNet)* for operators to deploy systems, where large macrocells give seamless service coverage while small picocells enhance signals in hotspots (e.g., stores or airports). To serve a growing number of UEs, operators prefer deploying many picocells in a macrocell to mitigate its load [3]. However, the density of UEs fluctuates over different times. For example, there are many UEs in a downtown office area on workdays, but it is nearly empty on weekends [4]. In this case, most picocell eNBs[2] are idle but still keep active. In fact, [5] shows that 80% energy of a wireless system is spent by eNBs. Thus, the key issue of green communications is to save energy of picocell eNBs.

The pricing policy, on the other hand, decides the amount of profit earned by an operator, which can be static or dynamic [6]. A static pricing policy asks users to pay a fixed rate for the service. It is easy for the operator to manage the billing mechanism but some users may overuse the service, causing network congestion and eroding the profit. A dynamic pricing policy charges each user depending on the amount of service actually consumed. In general, when the usage amount exceeds a threshold, the user will be asked to pay more money. Thus, the dynamic pricing policy allows the operator to increase its profit when the network load grows.

The issues of energy saving and dynamic pricing have great impact on resource scheduling. Many solutions view each cell as independent one and schedule resource cell by cell [2]. In a HetNet, it is better to schedule resource on a macrocell along with its picocells. Specifically, we can transfer the service of some picocell UEs to nearby cells and make idle eNBs sleep, thereby reducing energy consumption. Moreover, according to the *supply and demand theory* [7], the charge for a service has a large influence on its usage amount. Thus, when the network is saturated, we can request the users with large demands for higher fees to avoid them congesting the network, which facilities the resource scheduling process. However, how to integrate both energy saving and dynamic pricing with resource scheduling is not well addressed in the literature.

Therefore, we propose an *Energy-efficient Pricing and resource Scheduling (EPS)* framework that considers a common HetNet scenario in crowded cities [8], where picocells are densely deployed in each macrocell. Based on most pricing policies [6], we also divide users into gold, silver, and bronze classes, where high-class users pay more money but they are given precedence to use resource. Then, EPS contains three modules. The *eNB controlling module* divides picocell eNBs into groups, and selects one eNB as the *coordinator (CR)* to manage UEs in each group. When a picocell eNB has a heavy load, the CR transfers some of its UEs to other picocells in the group for load balance. If only few UEs are in a picocell, the CR asks nearby picocell eNBs in the same group to take over these UEs. Thus, the picocell eNB can be turned off to save energy. Then, the *scheduling module* adopts a two-layer scheduling strategy. The UE-based scheduler refers to the user class and channel quality of a UE to decide the number of

The authors are with the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, 80424, Taiwan. Email: ycwang@cse.nsysu.edu.tw; ooosss945@gmail.com.

[1] The case holds when UEs adopt SISO (single-input single-output) or SU-MIMO (single-user multiple-input multiple-out) techniques to receive data.

[2] In LTE-A, a base station is called an Evolved Node B, which is usually abbreviated as eNB.

PRBs given to it. The flow-based scheduler further allots these PRBs to the UE's flows by their delay demands to provide QoS to real-time flows. Finally, the *billing module* computes the fee charged to each user based on his/her class and usage amount of resource. It offers both peak and off-peak rates depending on the network load. So, we can increase service utilization and alleviate congestion, which improves network throughput.

The contribution of this paper is to manage resource efficiently in an LTE-A system by addressing both energy saving and dynamic pricing. In the literature, a few studies (e.g., our previous work in [9]) also jointly consider resource scheduling and pricing policy. However, there are some differences between our EPS framework and existing solutions. First, while most solutions schedule resource in a single cell, EPS aims at the *HetNet scenario* by grouping picocells and asking each CR to arrange UEs in its group. Thus, EPS can balance loads of picocells and also let idle eNBs sleep to save energy. Second, EPS gives precedence to *urgent flows* to get resource to reduce their packet dropping. Moreover, it fairly allots PRBs to non-urgent flows (by their demands) to avoid some flows starvation. These designs are peculiar to EPS. Third, EPS uses a peak rate to mitigate network congestion and gives an off-peak rate to attract people using resource when the network load is light. This idea of *differentiating charges* does not appear in existing solutions. Through simulations, we show that EPS outperforms other methods in terms of operator profit, network throughput, and energy consumption of eNBs.

We outline the paper as follows: Section 2 sketches LTE-A and Section 3 surveys related work. We present our network model in Section 4, followed by the design and discussion of EPS in Section 5. Section 6 then gives simulation study. Finally, we conclude the paper in Section 7.

2 Overview of LTE-A Systems

An LTE-A system can be split up into back and front ends. The back end is a *core network* that handles jobs of system management such as connecting to exterior networks (e.g., Internet), finding routing paths, sending control signals, and supporting UEs' mobility. The *policy and charging rules function (PCRF)* plays a key role in policy enforcement and user charging [10]. It refers to the application function to check if a UE obeys the subscription to transmit data. Besides, PCRF provides offline and online charging mechanisms. The offline mechanism gathers charging statistics in each session. When the usage amount of a UE reaches the upper limit, the online mechanism allows PCRF to cut off its service.

The front end contains *cells* controlled by eNBs, which can exchange information with X2 interfaces. Each eNB allocates resource based on PRBs, where a PRB has 0.5ms duration and 180kHz bandwidth. LTE-A divides time into *transmission time intervals (TTIs)*, and there are 6, 15, 25, 50, 75, and 100 PRBs available in a TTI (=1ms) when the channel's bandwidth is 1.4, 3, 5, 10, 15, and 20MHz, respectively. Each UE renders its channel condition via a *channel quality indicator (CQI)* to the eNB to decide the *modulation and coding scheme (MCS)* for the PRBs [11]. When a UE has a larger CQI (i.e., better channel quality), it can use PRBs with a more complex MCS that transmit more data bits. Otherwise, a simpler MCS is used to provide robust data transmission in a lower speed.

To define the QoS demand of a flow, LTE-A uses *QoS class identifier (QCI)* that includes its delay budget and loss rate [12]. The delay budget gives the maximum tolerable latency of packet transmission. When the delay of a packet overtakes the budget, it is dropped due to invalidity. The loss rate is the maximum tolerable ratio of dropped packets to total packets of the flow. With QCI, LTE-A divides flows into *guaranteed-bit-rate (GBR)* and *non-GBR* groups. GBR flows support real-time services with stringent delay demands like VoIP and live-streaming video. TCP-based services with loose deadlines are supported by non-GBR flows. Thus, GBR flows could have smaller QCIs and delay budgets than non-GBR flows.

3 Literature Survey

In this section, we survey LTE-A scheduling methods and pricing policies. Then, we discuss the studies which jointly consider resource scheduling and pricing. Finally, we present existing solutions to save energy of eNBs.

3.1 Scheduling Methods in LTE-A

There are a few basic scheduling methods in LTE-A [2]. MT (maximum throughput) serves UEs in sequence of their channel quality. PF (proportional fair) addresses user fairness [13] by selecting a UE with the maximum $r_i/\overline{r}_i$ value to get PRBs, where r_i and $\overline{r}_i$ are the UE's current and past data rates, respectively. M-LWDF (modified largest weighted delay first) adds a weight w_i and packet delay d_i to PF to support real-time services. EXP-PF (exponential proportional fair) adopts an equation of $\exp((w_i d_i - d_M)/(1+\sqrt{d_M}))$ to improve PF, where d_M is the mean packet delay. LOG-RULE and EXP-RULE consider spectral efficiency s_i of each UE. They give each PRB to a UE with the largest $(s_i \cdot \log f_1)$ and $(s_i \cdot \exp f_2)$ values, where f_1 and f_2 are two functions defined by LOG-RULE and EXP-RULE, respectively.

Some methods are devoted to increasing LTE-A throughput. Both [14], [15] use utility functions to find satisfaction degrees of UEs by their throughput. Then, UEs bid for PRBs via utility values. The work [16] uses a virtual queue to predict packet arrival, and removes the packets that will miss deadlines to save bandwidth. It then uses MT to allocate PRBs to improve throughput. The study [17] divides UEs into good-, average-, and poor-channel groups, and picks UEs from each group to get resource to raise throughput and avoid starvation of UEs.

The fairness issue is also discussed. The work [18] uses a bankruptcy game to fairly allot resource to UEs, and mends the scheduling result by EXP-RULE to raise throughput. By the Nash's solution, [19] derives a fairness criterion for eNBs to allocate PRBs, whose goal is to make the result be Pareto-optimal [20]. The study [21] computes the difference between the amount of data actually received by a UE and the amount of resource that it expects to get. With the difference, a credit value is derived to decide the priority of each UE to get PRBs for the purpose of keeping fairness.

Some studies aim at reducing packet delay. Liu et al. [22] modify PF by first sending packets with the earliest deadline. The study [23] divides flows into urgent and non-urgent ones. Urgent flows are GBR flows whose packets will be dropped soon, so they can get PRBs with a top priority. The work [24] uses MT to find the number of PRBs given to UEs. It then makes non-urgent flows return some PRBs, and reassigns them to urgent flows to reduce packet dropping. In [25], a cooperative game is used to allot PRBs, whose goal is to lower dissatisfaction of GBR flows in terms of packet dropping.

A few studies handle video transmissions. In [26], the eNB decides resource allocation along with coding of a video flow based on its data rate, delay, and distortion to play it smoothly. The work [27] estimates the amount of data that a video flow should send to support QoS, and uses PF to allocate PRBs to the flow. The study [28] develops a scheduling method to reduce delays of video flows and also a handoff method to keep service continuity when a user moves between cells. We can observe that none of the above studies consider integrating resource scheduling with dynamic pricing and energy saving.

3.2 Pricing Policies in Cellular Systems

There have been various pricing policies developed in GSM, GPRS, and UMTS systems[3] [29]. In *metered charging*, each user pays a fee for network connection and also a fee based on the service time. In *fixed price charging* (also called *flat-rate pricing*), users are charged for a fixed rental rate despite their usage amount of service. *Packet charging* counts packets sent in each session and charges the user accordingly. In *expected capacity charging*, the operator charges users by a usage profile, which gives the amount of capacity that they expect to get. It requires a network filter to tag excess traffic. In *Paris-metro charging*, users select their preferred classes with different prices and service quality, just like travel classes in public transport systems. *Market-based reservation charging* is an auction-based method, where the operator refers to the preference profile and bid of each user to route packets.

As LTE-A materializes its resource by PRBs, many pricing policies for LTE-A charge users based on the number of PRBs consumed, and they usually classify users into gold, silver, and bronze. The *static pricing policy* [30] computes the fee charged to a user with class ξ_i by $M_i = C_{\text{P}}(\xi_i) \cdot n_i$, where $C_{\text{P}}(\xi_i)$ is the cost per PRB depending on ξ_i and n_i is the number of PRBs spent by the user. The *network-load based pricing (NLP) policy* [31] adjusts the price based on the network load L, which charges a user with QoS level x by

$$M_i = [C_{\text{V}}(\xi_i) \times (\hat{e} - \hat{e}^{-\alpha x})L] \times n_i, \tag{1}$$

where $\hat{e} \approx 2.71828$ is the Euler's number, α is a coefficient, and $C_{\text{V}}(\xi_i)$ is a variable charge based on a threshold δ. When $L > \delta$, $C_{\text{V}}(\xi_i)$ changes with ξ_i, so different classes of users are charged diversely to raise operator profit when the network load is heavy. Otherwise, users are charged fairly by setting $C_{\text{V}}(\xi_i)$ to a constant C_{F}. The *subscriber class based pricing (SCP) policy* [32] works the same with the static pricing policy as $L \leq \delta$. Otherwise, it asks users to pay different fees:

$$\text{gold users:} \quad M_i = [C_{\text{P}}(\xi_{\text{G}}) + C_{\text{E}}] \times n_i, \tag{2}$$

$$\text{silver users:} \quad M_i = [2C_{\text{P}}(\xi_{\text{G}}) + C_{\text{E}}] \times n_i, \tag{3}$$

$$\text{bronze users:} \quad M_i = [2C_{\text{P}}(\xi_{\text{G}}) + C_{\text{P}}(\xi_{\text{S}}) + C_{\text{E}}] \times n_i, \tag{4}$$

where ξ_{G} and ξ_{S} are gold and silver classes, respectively. Here, $C_{\text{E}} = \beta/(n_{\text{A}} - n_{\text{G}})$ is an extra charge, where β is a constant, n_{A} is the number of total PRBs, and n_{G} is the number of PRBs reserved for gold users. Obviously, the above pricing policies ask users to pay more money to increase operator profit under a heavy network load. However, they do not lower the price to attract people using more resource as the network has a light load. It motivates us to propose off-peak rates in our billing module, which distinguishes EPS from existing work.

3.3 Joint Consideration of Scheduling and Pricing

A number of studies jointly address the issues of resource scheduling and pricing. The work [33] assumes that an eNB provides multiple carriers, where it has a price to use each carrier per unit bandwidth. Each UE selects a carrier to be its primary carrier and others will be secondary carriers. Then, the problem is how to allocate resource to UEs with carrier aggregation to maximize throughput, such that the charge for their allocated resource will be the minimum. Obviously, [33] deals with a different problem with ours.

In [34], a gradient-based scheme is used to allocate resource to UEs. When a UE is served, its gradient reduces and vice versa. The objective is to make all UEs have some common gradient (i.e., keeping user fairness). Thus, the eNB computes a load metric for each UE based on the PF method. In addition, to fairly charge users, the load metric of each UE is scaled by a weight that depends on its user class (e.g., gold, silver, and bronze). However, [34] aims to charge users based on their usage amount of resource in a fair manner, instead of increasing operator profit and system utilization. Therefore, it has a different objective with our EPS framework.

The study [35] proposes an auction-based method for resource allocation and pricing. Each UE has a utility function to depict its QoS demand, and it sends a bid to the eNB to ask for the desired rate (based on the utility value). Then, the eNB replies a shadow price to the UE to make it revise the bid. The procedure is repeated until the difference between two bids is below a threshold. Then, the eNB grants the UE's rate and its user has to pay the shadow price. However, this method requires many rounds of negotiations for pricing between every UE and the eNB, which incurs a high message overhead. Besides, [35] considers only one macrocell, and the result cannot be directly applied to an LTE-A HetNet.

Our previous work [9] proposes a *pricing-aware resource scheduling (PARS)* framework. It applies a price-based weight to MT and M-LWDF to allocate resource, so high-class UEs are given precedence to get PRBs. However, when a low-class UE has good channel quality, it can borrow PRBs from high-class ones to improve throughput. For the pricing policy, PARS adds an extra charge C_{E} to NLP, and modifies Eq. (1) by

$$M_i = [(\hat{e} - \hat{e}^{-y})L] \times [C_{\text{V}}(\xi_i) \times n_i + C_{\text{E}} \times n'_i], \tag{5}$$

where y is the QoS level, n_i is the number of PRBs given to a UE, n'_i is the number of extra PRBs that the UE borrows from others. In Eq. (5), PARS sets the QoS level y to the QCI index of a flow (referring to Table 3) and keeps y in $[1, 9]$. When the QCI index is above 9, y is directly set to 9. As discussed in Section 1, our EPS framework improves PARS by considering the HetNet scenario and saving energy of eNBs. Moreover, EPS favors urgent flows to reduce packet dropping and also offers differentiating charges to give flexibility in its pricing policy. Simulation results in Section 6 will show that EPS outperforms PARS on profit, throughput, and energy.

3.4 Energy Saving for eNBs

Various mechanisms are proposed to save energy spent by eNBs. The work [36] finds the sites to place small-cell eNBs to reduce the transmitted power of macrocell eNBs. With MIMO beam-forming and multiplexing gain, [37] develops a MAC protocol to save energy of eNBs. Jin et al. [38] apply cognitive radio to LTE-A, which allows eNBs to enter the power-saving state when no packets wait for transmission. In [39], each eNB

[3] GSM: global system for mobile communications, GPRS: general packet radio service, UMTS: universal mobile telecommunications system

TABLE 1: Summary of common notations.

notation	definition
U_j	set of UEs served by eNB e_j, where $\lvert U_j \rvert = N_j$
L_j, L	e_j's load and the network load
ζ_1, ζ_2	two lists of candidate UEs to be removed by e_j
r_i	data rate of UE u_i
ξ_i	user class of u_i (ξ_{G}: gold, ξ_{S}: silver, ξ_{B}: bronze)
$\gamma_{i,j}$	u_i's SINR from e_j
$V_{i,t}$	amount of GBR data that u_i should get in TTI t
$\varepsilon_{i,k}(t)$	flow $f_{i,k}$'s data to be sent in TTI t by FLS
$\rho_{i,k}(t)$	flow $f_{i,k}$'s queued data in TTI t
δ_{num}	a threshold for N_j
δ_{peak}, δ_{off}	two thresholds of using peak and off-peak rates
$\delta^{\mathrm{L}}_{\mathrm{load}}, \delta^{\mathrm{H}}_{\mathrm{load}}$	lower- and upper-bound thresholds for L_j
C_{E}, C_{F}	extra and constant charges for a PRB
C_{peak}, C_{off}	additional and bonus fees for a PRB
$C_{\mathrm{P}}(\xi_i)$, $C_{\mathrm{V}}(\xi_i)$	fixed and variable charges per PRB based on ξ_i
$\tilde{m}_{\mathrm{U}}$, $\tilde{m}_{\mathrm{E}}$, $\tilde{m}_{\mathrm{F}}$, $\tilde{m}_{\mathrm{P}}$	numbers of UEs, eNBs, flows, and PRBs
p_e	price elasticity
T	length of a period (in TTIs)

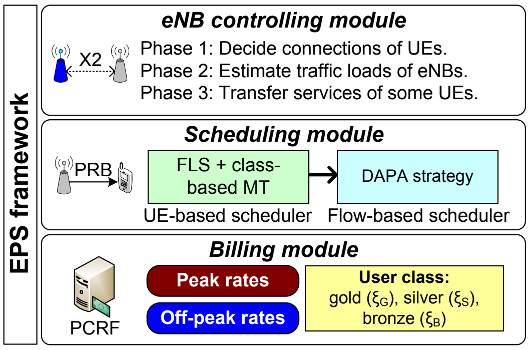


Fig. 1: The architecture of our EPS framework.

uses carrier aggregation to send data, and seeks to reduce its power on some subchannels for energy saving.

Discontinuous transmission (DTX) is a technique to manage energy expense of eNBs, which allows them sleeping to save energy [40]. The *sleep schedule* is critical, which decides when an eNB can sleep. The work [41] proposes a coordinated sleep schedule to minimize the duration when adjacent eNBs wake up to send data. So, it can reduce interference between cells. In [42], the sleep schedule is formulated by a traffic awareness problem. Given a set of traffic predictions by eNBs, the problem asks how to maximize successful traffic predictions (i.e., predicted traffic profile matches with actual traffic). It is NP-hard and a near-optimal solution by the game theory is proposed. The study [43] considers a HetNet where femtocells are deployed in a macrocell. It finds the sleep schedule of femtocell eNBs by their traffic loads, so as to keep the target throughput while saving energy of eNBs. The work [44] addresses a ping-pong effect on DTX, where eNBs may have on/off oscillations. It models user traffics by a Markov process, and adds a hysteresis time to the process to handle the effect. Comparing with prior work, our EPS framework organizes picocells into groups, and selects a CR to manage UEs in each group. It can actively transfer UEs among cells to balance loads of eNBs and also deactivate the eNBs serving just few UEs to avoid energy wastage.

4 Network Model

Let us consider an LTE-A HetNet where there are many picocells deployed in each macrocell. Since we seek to transfer UEs among picocells, it is natural to cluster picocell eNBs close to each other into the same group. To do so, we adopt the *enhanced agglomerative hierarchical clustering (eAHC)* scheme [45], which recursively groups picocell eNBs according to their positions. It contains three steps:

1. Initially, each picocell eNB e_j is viewed as a group g_j.
2. We pick two groups g_i and g_j whose *inter-group distance* $Z(g_i, g_j)$ is the minimum, which is the distance between two farthest eNBs $e_x \in g_i$ and $e_y \in g_j$. If $Z(g_i, g_j) \leq \delta_g$, we merge g_i and g_j together, where δ_g is a threshold to determine the diameter of a group.
3. Step 2 is repeated until no groups can be merged.

Since δ_g controls the size of each group, its value can depend on various parameters such as the propagation environment or the distribution of UEs. For example, if the environment allows better propagation of signals, we can set a larger δ_g value, as a picocell has a larger propagation range and it becomes easier to transfer UEs between nearby picocells. Moreover, in hotspots or urban areas, since the density of picocells may be very high, we could set a smaller δ_g value to avoid a group including too many picocells. Due to the page limit, we leave the investigation of δ_g in our future work. Here, we suggest setting δ_g as the maximum propagation distance of a picocell eNB, so the CR can transfer UEs among picocells in its group.

We define three modes to control the operations of eNBs. In the *active* mode, an eNB supports regular data transmission for its UEs. In the *sniff* mode, an eNB keeps synchronization with UEs by sending only control messages. In the *sleep* mode, an eNB turns off its wireless interface to UEs to save energy. In each group of picocell eNBs, we select one as the CR to manage UEs. Other eNBs report their cell statuses (e.g., loads and UE numbers) to the CR. Then, the CR sends commands to a member eNB to trigger events such as transferring its UEs to other cells or switching the eNB's mode. These commands are exchanged through the X2 interface, and the detail of message flows can be found in [46].

Since a CR takes charge of the management job in its group, it should keep active. Besides, when an eNB is busy in serving UEs, it is not a good candidate for CR. Based on the observations, we select the *active* eNB whose load is the lightest to be the CR in each group. Specifically, let L_j be the traffic load of an eNB. We select the CR in a group g_i by

$$\arg\min_{e_j \in g_i} \{L_j \mid L_j \geq \delta^{\mathrm{L}}_{\mathrm{load}}\}, \tag{6}$$

where $\delta^{\mathrm{L}}_{\mathrm{load}}$ is a lower-bound threshold on picocell load (as discussed later in Section 5.1.3). In case that no eNB can be found from Eq. (6) (i.e., the case occurs when each eNB in g_i has a load below $\delta^{\mathrm{L}}_{\mathrm{load}}$), we select the eNB whose load is the largest to be the CR.

Without loss of generality, UEs are divided into classes of gold (ξ_{G}), silver (ξ_{S}), and bronze (ξ_{B}). The class of each UE decides its priority to get resource, where $\xi_{\mathrm{G}} > \xi_{\mathrm{S}} > \xi_{\mathrm{B}}$. In addition, PCRF measures the amount of resource spent by UEs to charge their users, where gold, silver, and bronze users will pay high, medium, and low unit prices, respectively. Some pricing policies in Section 3.2 put a restriction on flows that each UE can use (e.g., [32] prohibits bronze UEs from using GBR flows). We relax this assumption for flexibility. Table 1 summarizes our notations.

5 The Proposed EPS Framework

EPS has three modules shown in Fig. 1. The eNB controlling module arranges UEs in each group of picocells. It decides initial connections of UEs and estimates the amount of resource spent in a cell. Then, the CR alleviates congestion in the picocells with heavy loads by transferring their UEs. When an eNB serves only few UEs, the CR allows the eNB to sleep by asking others to take over its UEs. In the module, the CR exchanges information with other eNBs via X2 interfaces.

The scheduling module helps each eNB assign PRBs to its UEs, which contains two schedulers. The *UE-based scheduler* combines the frame-layer scheduling (FLS) strategy [27] with a class-based MT method to find the number of PRBs given to each UE, which refers to its GBR demand, channel quality, and user class. Then, the *flow-based scheduler* further allots the PRBs acquired by a UE to its flows based on their delays and demands, so as to reduce packet dropping.

Our pricing policy is implemented by the billing module in PCRF. When the network is saturated, we adopt peak rates to make users pay more, so as to increase operator profit and avoid them overly consuming resource. When the network has a light load, off-peak rates are applied to let users pay less, so as to encourage them increasing demands to raise service utilization. In the billing module, we will take user classes and service types (i.e., QCIs of flows) into consideration.

For ease of management, we divide time into *periods* of T TTIs, and conduct EPS in every period. Below, we detail our design in each module, followed by a discussion of EPS.

5.1 Design of the eNB Controlling Module

We use two parameters to decide the mode of an eNB e_j: 1) traffic load L_j and 2) the number of served UEs N_j. Then, this module contains three phases. In phase 1, each UE initially associates with an eNB based on its user class and signal to interference plus noise ratio (SINR). By the association, phase 2 estimates the amount of resource spent in a cell. Then, phase 3 arranges UEs in each group to balance loads of picocell eNBs and also turn off unused eNBs to save energy.

5.1.1 Phase 1–Decide Connections of UEs

In general, a UE prefers linking to the eNB that provides the highest SINR, so it can raise the data rate by using a complex MCS. Moreover, as picocells are often used to enhance signal strength in small regions, they could provide better service quality than the background macrocell. Since high-class users pay more money, we can help their UEs connect to picocell eNBs with good SINRs to support QoS. Therefore, we decide the initial connection of each UE u_i as follows:

- If $\xi_i = \xi_{\mathrm{B}}$ (i.e., a bronze UE), we always let u_i associate with the macrocell eNB.
- If $\xi_i = \xi_{\mathrm{S}}$ (i.e., a silver UE), we compute u_i's SINR with each eNB e_j by

$$\gamma_{i,j} = \frac{\hat{p}(u_i, e_j)}{I_{\text{noise}} + \sum_{e_k \in \mathcal{E}, e_k \neq e_j} \hat{I}(u_i, e_k)}, \tag{7}$$

where $\hat{p}(u_i, e_j)$ is u_i's received power from e_j, I_{noise} is interference from the environmental noise[4], $\hat{I}(u_i, e_k)$ is interference from an eNB e_k, and $\mathcal{E}$ is the set of eNBs. Then, u_i connects to the eNB with the largest $\gamma_{i,j}$ value.

TABLE 2: Required SINR [51] and MCS of each CQI.

CQI	SINR	MCS	CQI	SINR	MCS
1	-6.936 dB	QPSK	9	8.573 dB	16QAM
2	-5.147 dB	QPSK	10	10.366 dB	64QAM
3	-3.180 dB	QPSK	11	12.289 dB	64QAM
4	-1.253 dB	QPSK	12	14.173 dB	64QAM
5	0.761 dB	QPSK	13	15.888 dB	64QAM
6	2.699 dB	QPSK	14	17.814 dB	64QAM
7	4.694 dB	16QAM	15	19.829 dB	64QAM
8	6.525 dB	16QAM			

- If $\xi_i = \xi_{\mathrm{G}}$ (i.e., a gold UE), we also use Eq. (7) to find its SINR from each eNB e_j. However, if e_j is a picocell eNB, we add a small bias φ to SINR $\gamma_{i,j}$ to increase the possibility that u_i can associate with a picocell eNB.

Our design borrows the notion of traffic offloading [47], [48] and cell range expansion [49], [50] in HetNets. However, different from these methods, we let each UE join a cell by referring to its SINR and also user class. Thus, we can differentiate between UEs by eNB association based on their classes and provide different QoS support to them accordingly.

5.1.2 Phase 2–Estimate Traffic Loads of eNBs

Once deciding the UEs in a cell, we can estimate the traffic load of its eNB (in terms of PRBs consumed). Given SINR of a UE, we find its CQI by Table 2. Based on CQI, the LTE-A standard [52] gives three tables to compute the number of data bits to be sent to the UE in a TTI. The *CQI-MCS translation table* maps the CQI value to an MCS index. Then, the *MCS-TBS translation table* uses the MCS index to find the index of TBS (transport block size), which indicates the number of data bits carried by one PRB. Finally, given the TBS index and the number of PRBs, the *TBS-bit translation table* returns the number of data bits that the UE can receive from these PRBs. For convenience, we use a function $f_{\mathbf{T}}(\gamma_{i,j}, n_{i,j})$ to denote the translation of three tables along with Table 2, where $n_{i,j}$ is the number of PRBs that eNB e_j gives to a UE u_i. Suppose that u_i has a demand of data rate r_i (in bps). Then, we can calculate the minimum number of PRBs to satisfy its demand as follows:

$$n_{i,j}^{\min} = \arg\min_{n_{i,j}} \left\{ f_{\mathbf{T}}(\gamma_{i,j}, n_{i,j}) \times T \geq \frac{r_i T}{1000} \right\}. \tag{8}$$

The term $f_{\mathbf{T}}(\gamma_{i,j}, n_{i,j}) \times T$ gives the number of data bits that these $n_{i,j}$ PRBs can carry, while the term $\frac{r_i T}{1000}$ indicates the number of data bits that u_i expects to receive. With Eq. (8), we calculate the traffic load of an eNB e_j by

$$L_j = \frac{\sum_{u_i \in U_j} n_{i,j}^{\min}}{\tilde{n}_j T}, \tag{9}$$

where U_j is the set of UEs served by e_j and $\tilde{n}_j$ is the number of PRBs offered by e_j in a TTI (referring to Section 2).

5.1.3 Phase 3–Transfer Services of Some UEs

In each group, only *active* eNBs report their L_j and N_j parameters to the CR. For each active eNB e_j, we decide its mode by three cases[5]:

- $L_j \geq \delta_{\text{load}}^{\mathrm{L}}$: We let e_j keep active, as its load is above the lower-bound threshold. If $L_j > 1$ (i.e., it does not have enough resource to serve all UEs), e_j sends an *load-balance request* to the CR for UE arrangement.

[4] The noise is a Gaussian white noise. We can compute its value by taking the product of noise figure, spectral density, and channel bandwidth.

[5] We can set $\delta_{\text{load}}^{\mathrm{L}} \geq 1/4$ and δ_{num} to 1/3 of the average number of UEs in each picocell.

- $L_j < \delta^{\mathrm{L}}_{\mathrm{load}}$ and $N_j < \delta_{\mathrm{num}}$: Since e_j has a pretty light load and serves just few UEs (i.e., below the threshold δ_{num}), it sends a *sleep request* to the CR.
- $L_j < \delta^{\mathrm{L}}_{\mathrm{load}}$ and $N_j \geq \delta_{\mathrm{num}}$: As e_j is nearly idle but there are more UEs to be served (i.e., each UE has a small demand), it then changes to the sniff mode.

Note that we do not make an eNB sleep in case 3, since some UEs may raise their demands later (and cause loss of service). Besides, these sniffing eNBs can help share loads of nearby eNBs. The sniff mode is *temporary*. After the CR arranges UEs, these eNBs will switch to the active or sleep modes.

In case 1, when e_j overloads (i.e., $L_j > 1$), it decides two lists of candidate UEs ζ_1 and ζ_2 to be removed from its cell. List ζ_1 gives a set of UEs such that after e_j removes them, it can decrease the load to $L_j \leq \delta^{\mathrm{H}}_{\mathrm{load}}$, where $\delta^{\mathrm{H}}_{\mathrm{load}} > 1$ is an upper-bound threshold on picocell load (e.g., we can set $\delta^{\mathrm{H}}_{\mathrm{load}}$ to 5/4). List ζ_2 includes a set of UEs such that if e_j *further* removes them, it can decrease the load to $L_j \leq 1$. Note that when $L_j < \delta^{\mathrm{H}}_{\mathrm{load}}$, e_j generates only list ζ_2. We then use two rules for e_j to iteratively find out candidate UEs: 1) Select the UE with the lowest user class. 2) If there are multiple choices, the UE with minimum CQI will be selected. Then, e_j appends ζ_1 and ζ_2 to its load-balance request. Remark 1 discusses the reason why we adopt the threshold $\delta^{\mathrm{H}}_{\mathrm{load}}$.

On getting parameters and requests from member eNBs, the CR first handles overloading eNBs. For each UE u_i in list ζ_1 of a load-balance request, the CR uses four rules *in sequence* to choose an eNB to take over u_i:

R1. If u_i is covered by an active eNB e_a such that $L_a \leq 1$ after e_a serves u_i, e_a can take over u_i. If there is a tie, we pick the eNB with the *smallest* L_a value to serve u_i.

R2. If u_i is covered by a sniffing eNB e_s such that $L_s \leq 1$ after e_s serves u_i, e_s can take over u_i. If there is a tie, we pick the eNB with the *largest* L_s value to serve u_i.

R3. If u_i is covered by a sleeping eNB e_z, it is added to a pending list ζ_p of e_z. In case that u_i is covered by multiple sleeping eNBs, we choose the eNB whose pending list contains the most number of UEs.

R4. The macrocell eNB will take over u_i.

For each sleeping eNB e_z whose pending list ζ_p is non-empty, if $L_z \geq \delta^{\mathrm{L}}_{\mathrm{load}}$ when e_z serves all UEs in ζ_p, the CR will send e_z an *awaking command* along with ζ_p to wake it up to serve these UEs. Otherwise, the CR asks the macrocell eNB to take over the UEs in ζ_p.

The CR then uses rules R1 and R2 to transfer each UE in list ζ_2 to other eNBs. In case that ζ_2 is not empty but no eNB can take over the UEs in ζ_2, the CR notifies the eNB which sent the load-balance request of the residual UEs in ζ_2, so the eNB should still serve these UEs.

Afterwards, for each eNB which sent a sleep request, the CR uses rules R1, R2, and R4 to transfer all of its UEs to other cells, and sends a *sleeping command* to turn off the eNB. Note that the sleeping eNB will wake up on the next period.

The CR finally handles sniffing eNBs. For each sniffing eNB that need not take over any new UE, the CR transfers all of its UEs to other cells and lets it sleep. Then, the CR sends awaking commands to residual sniffing eNBs to make them active, which includes a list of new UEs that each eNB should serve. Remark 2 discusses the idea behind these transfer rules. Lemma 1 then analyzes the computation complexity of the eNB controlling module.

Remark 1 (Threshold $\delta^{\mathrm{H}}_{\mathrm{load}}$)**.** The idea behind using $\delta^{\mathrm{H}}_{\mathrm{load}}$ is to distinguish *excessively* overloaded eNBs (i.e., $L_j \geq \delta^{\mathrm{H}}_{\mathrm{load}}$) from *slightly* overloaded eNBs (i.e., $1 < L_j < \delta^{\mathrm{H}}_{\mathrm{load}}$). When an eNB e_j is excessively overloaded, we should ask sleeping eNBs or the macrocell eNB to help offload its traffic (if rules R1 and R2 fail), so as to alleviate serious congestion in e_j's cell. However, when e_j is slightly overloaded, it is not economic to wake up a sleeping eNB to share e_j's load, since the sleeping eNB may only need to serve just few UEs. Similarly, it is also uneconomic to ask the macrocell eNB to take over e_j's UEs, as the macrocell eNB may be burdened with a heavy load (if many slightly overloaded eNBs ask it to do so). In fact, when an eNB is slightly overloaded, we can use the scheduling module in Section 5.2 to efficiently distribute PRBs among UEs in its cell and still support QoS for GBR flows.

Remark 2 (Transfer rules)**.** Rule R1 borrows the notion of *packet fair queuing* [53], which always picks the eNB with minimum load to take over a new UE. Thus, we can ensure that each active eNB in a group can have a similar load in long term (i.e., achieving load balance). On the other hand, rule R2 prefers using fewer sniffing eNBs to take over UEs, so as to make more sniffing eNBs sleep to save energy. That is why we choose a sniffing eNB with maximum load in rule R2. Then, rule R3 is used to wake up some sleeping eNBs to help offload the traffic of those active eNBs that have excessive loads (i.e., $L_j \geq \delta^{\mathrm{H}}_{\mathrm{load}}$). Since it is not energy-efficient to wake up a sleeping eNB in order to serve UEs with just small demands, the CR only sends awaking commands to those sleeping eNBs whose pending lists satisfy the condition of $L_z \geq \delta^{\mathrm{L}}_{\mathrm{load}}$. Obviously, rule R3 will not be used to transfer UEs of an eNB that will go to sleep. Finally, rule R4 is to cope with the cases when nearby picocell eNBs cannot cover the UEs of an eNB e_j or they do not have enough resource to share e_j's load. Thus, we have to ask the macrocell eNB to take over e_j's UEs. This rule will not be used to transfer UEs in list ζ_2, as discussed in Remark 1.

Lemma 1. *Given $\tilde{m}_{\mathrm{U}}$ UEs and $\tilde{m}_{\mathrm{E}}$ eNBs, the eNB controlling module takes time of $O(\tilde{m}_{\mathrm{U}}\tilde{m}_{\mathrm{E}})$ in the worst case.*

Proof: Phase 1 decides initial connections of UEs. When a UE is bronze, it is assigned to the macrocell. Otherwise, we find the UE's SINR from each eNB by Eq. (7). The worst case occurs when there are no bronze UEs, so phase 1 spends $O(\tilde{m}_{\mathrm{U}}\tilde{m}_{\mathrm{E}})$ time. Then, phase 2 uses Eq. (8) to find the number $n^{\mathrm{min}}_{i,j}$ of PRBs to meet a UE's demand, where we use four tables (i.e., Table 2 and three LTE-A tables) to compute $f_{\mathbf{T}}(\gamma_{i,j}, n_{i,j})$. Each table-finding operation takes constant time. Then, we estimate load L_j of each eNB by Eq. (9). As each UE links to one eNB and there are $\tilde{m}_{\mathrm{U}}$ UEs, phase 2 will check every UE once in Eq. (9). Thus, this phase spends $O(5\tilde{m}_{\mathrm{U}})$ time. Finally, phase 3 uses three cases to decide the mode of each eNB (by checking its L_j and N_j values), which spends $O(\tilde{m}_{\mathrm{E}})$ time. It takes $O(\tilde{m}_{\mathrm{U}})$ time to find lists ζ_1 and ζ_2 of eNBs by checking each UE. For the transfer rules, the worst case occurs when only rules R1 or R2 are adopted. We can use both minimum and maximum heaps for the two rules. It takes $O(\tilde{m}_{\mathrm{E}})$ time to build a heap. When selecting an eNB by rules R1 or R2, we conduct one deletion and one insertion on the heap, where each operation takes $O(\lg \tilde{m}_{\mathrm{E}})$ time. As each UE is transferred at most once, the arrangement of UEs spends time of $2O(\tilde{m}_{\mathrm{E}}) + 2O(\lg \tilde{m}_{\mathrm{E}}) \times \tilde{m}_{\mathrm{U}} = O(\tilde{m}_{\mathrm{U}} \lg \tilde{m}_{\mathrm{E}})$. Thus, the total complexity is $O(\tilde{m}_{\mathrm{U}}\tilde{m}_{\mathrm{E}}) + O(5\tilde{m}_{\mathrm{U}}) + O(\tilde{m}_{\mathrm{E}}) + O(\tilde{m}_{\mathrm{U}}) +$

$O(\tilde{m}_{\mathrm{U}} \lg \tilde{m}_{\mathrm{E}}) = O(\tilde{m}_{\mathrm{U}} \tilde{m}_{\mathrm{E}})$. □

5.2 Design of the Scheduling Module

After arranging UEs by the above module, if an eNB e_j has load of $L_j \leq 1$, it means that e_j has enough resource to meet each UE's demand. Thus, we directly use the MT method discussed in Section 3.1 for e_j to allocate PRBs to its UEs.

Otherwise, each eNB follows three guidelines to schedule resource: 1) GBR flows are served first to support QoS. 2) UEs are prioritized by their user classes to get resource, where $\xi_{\mathrm{G}} > \xi_{\mathrm{S}} > \xi_{\mathrm{B}}$. 3) The UEs with good channel quality are given precedence over others for resource allocation to raise throughput. Thus, we propose a two-layer scheduling strategy for the eNB to allot PRBs to GBR flows in its cell. As shown in Fig. 1, the UE-based scheduler combines the FLS and class-based MT methods to decide the number of PRBs given to each UE that has GBR flows. The flow-based scheduler further deals out these PRBs to the UE's GBR flows based on their packet delays and traffic demands. Then, if the eNB still has PRBs left, the eNB allocates them to non-GBR flows by the class-based MT method. Below, we detail the two schedulers.

5.2.1 UE-based Scheduler

Given a GBR flow $f_{i,k}$ of UE u_i, we use FLS to estimate the amount $\varepsilon_{i,k}(t)$ of $f_{i,k}$'s data that u_i should get in TTI t to support QoS. Let us denote by $\rho_{i,k}(t)$ the amount of $f_{i,k}$'s data queued by the eNB in TTI t. Then, we can compute the amount of $f_{i,k}$'s queued data in TTI $(t+1)$ by $\rho_{i,k}(t+1) = \rho_{i,k}(t) + \lambda_{i,k}(t) - \varepsilon_{i,k}(t)$, where $\lambda_{i,k}(t)$ is the amount of data produced by $f_{i,k}$ in TTI t. After a bit of algebra, we can derive that

$$\varepsilon_{i,k}(t) = \lambda_{i,k}(t) + \rho_{i,k}(t) - \rho_{i,k}(t+1). \tag{10}$$

To keep the *bounded-input, bounded-output (BIBO) stability*[6] of u_i's queue, [27] defines a control law for $\rho_{i,k}(t)$ by

$$\rho_{i,k}(t) = \mu_{i,k}(t) * \lambda_{i,k}(t), \tag{11}$$

where '$*$' is a discrete-time convolution and

$$\mu_{i,k}(t) = \sum\nolimits_{x=0}^{T} c_{i,k}(x)\phi(t-x). \tag{12}$$

In Eq. (12), $\phi(t-x)$ is the Kronecker pulse [54], whose value is either 0 or 1, and $c_{i,k}(x)$ should meet two conditions:

$$\begin{aligned} 0 \leq &c_{i,k}(x) \leq 1 \quad x = 0, 1, 2, \cdots, \\ &c_{i,k}(x) \geq c_{i,k}(x+1), x \geq 1 \text{ with } c_{i,k}(x) \in \mathbb{R}. \end{aligned} \tag{13}$$

By applying Eq. (12) to Eq. (11), we obtain that

$$\rho_{i,k}(t) = \sum\nolimits_{x=0}^{T} c_{i,k}(x)\lambda_{i,k}(t-x). \tag{14}$$

By combining Eqs. (10) and (14), we finally derive that

$$\begin{aligned} \varepsilon_{i,k}(t) = &\lambda_{i,k}(t) + \sum\nolimits_{x=0}^{T} c_{i,k}(x)\lambda_{i,k}(t-x) - \\ &\sum\nolimits_{x=1}^{T} c_{i,k}(x)\lambda_{i,k}(t+1-x). \end{aligned} \tag{15}$$

In EPS, we set $c_{i,k}(0) = 0$ and $c_{i,k}(x) = \frac{1}{2^{x-1}}$, where $x \geq 1$, to satisfy both conditions in Eq. (13). Specifically, it is clear that $0 \leq \frac{1}{2^{x-1}} \leq 1$, for all $x \geq 1$. Besides, we can derive that

$$c_{i,k}(x) - c_{i,k}(x+1) = \frac{1}{2^{x-1}} - \frac{1}{2^{(x+1)-1}} = \frac{1}{2^x} > 0.$$

Therefore, the correctness of our setting for $c_{i,k}(x)$ is justified. Then, the amount of GBR data that UE u_i expects to receive in TTI t can be calculated by

$$V_{i,t} = \sum \{\varepsilon_{i,k}(t) \mid \forall f_{i,k} \text{ of } u_i\}. \tag{16}$$

However, an eNB e_j may not have enough resource to send out the $V_{i,t}$ amount of GBR data for every UE in its cell. Thus, we develop a *class-based MT method* for UEs to bid for PRBs, which considers their user classes and channel quality. Let r_i be the data rate of a PRB for UE u_i (depending on its CQI, which is discussed in Section 5.1.2). Then, e_j uses Eq. (17) to select a UE (with GBR flows) to get each PRB:

$$u_i = \arg\max\nolimits_{u_i \in U_j} (\hat{w}_i^{\mathrm{C}} \times r_i). \tag{17}$$

Here, $\hat{w}_i^{\mathrm{C}}$ is a weight based on u_i's class, which is defined by

$$\hat{w}_i^{\mathrm{C}} = \frac{C_{\mathrm{V}}(\xi_i)}{C_{\mathrm{V}}(\xi_{\mathrm{G}}) + C_{\mathrm{V}}(\xi_{\mathrm{S}}) + C_{\mathrm{V}}(\xi_{\mathrm{B}})}, \tag{18}$$

where $\xi_i \in \{\xi_{\mathrm{G}}, \xi_{\mathrm{S}}, \xi_{\mathrm{B}}\}$. When a UE obtains enough PRBs to satisfy its $V_{i,t}$ amount of GBR demand, it is removed from U_j in Eq. (17) to avoid getting too much resource. The above procedure is repeated until $U_j = \emptyset$ or e_j runs out of PRBs. We remark that since $C_{\mathrm{V}}(\xi_{\mathrm{G}}) > C_{\mathrm{V}}(\xi_{\mathrm{S}}) > C_{\mathrm{V}}(\xi_{\mathrm{B}})$, a high-class UE can increase its opportunity to get PRBs by using a large weight $\hat{w}_i^{\mathrm{C}}$ in Eq. (17), and vice versa.

5.2.2 Flow-based Scheduler

Suppose that a UE u_i is given n_i^{R} PRBs by the above scheduler and it has a set F_i of GBR flows. Then, this scheduler aims to allot the n_i^{R} PRBs to all flows in F_i. To do so, we propose a *delay-aware proportional allocation (DAPA)* strategy with three steps below.

Step 1: We find out *urgent flows* from F_i and let them get PRBs first. Here, a flow is urgent if its packets will expire and be dropped in the next TTI. Thus, we iteratively give a PRB to each urgent flow by round robin and deduct one from n_i^{R}, until no urgent flow will drop its packets. If $n_i^{\mathrm{R}} = 0$, DAPA terminates as all PRBs have been allocated.

Step 2: We then compute a *priority* for each flow in F_i. In M-LWDF [2], UE u_i is assigned with a priority to get PRBs:

$$p_i = -\frac{d_i \log \alpha_i}{\tau_i} \times \frac{r_i}{\overline{r}_i}, \tag{19}$$

where d_i is the head-of-line packet delay of u_i, and α_i and τ_i are u_i's packet loss rate and delay budget defined by it QCI (referring to Section 2), respectively. However, since every flow of u_i has the same values of r_i and $\overline{r}_i$ (i.e., u_i's current and past data rates, respectively), we thus compute the priority of a flow $f_{i,k} \in F_i$ by

$$p_{i,k} = (-d_{i,k} \log \alpha_{i,k})/\tau_{i,k}. \tag{20}$$

For a urgent flow, its $d_{i,k}$ value will be the delay of the first packet that will not expire in the next TTI.

Step 3: We finally allocate PRBs to all flows in F_i proportionally to their demands, and adjust the allocation result by their priorities. Let us denote by $m_{i,k}$ the number of PRBs

[6] If a system is BIBO stable [54], its output must be bounded in amplitude by giving a bounded input. Since the demands of flows are finite (i.e., bounded input), the eNB will not consume an infinite bandwidth (i.e., bounded output).

required to satisfy $f_{i,k}$'s demand, which can be calculated by the method in Section 5.1.2. Then, for each flow in F_i, it will be allocated with a number $n_{i,k}$ of PRBs:

$$n_{i,k} = \left\lceil \frac{m_{i,k}}{\sum_{\forall f_{i,h} \in F_i} m_{i,h}} \times n_i^{\mathrm{R}} \right\rceil. \tag{21}$$

However, it is possible that $\sum_{\forall f_{i,k} \in F_i} n_{i,k} > n_i^{\mathrm{R}}$. In this case, we sort all flows in F_i by their priorities (from low to high), and ask each flow to give up one PRB by round robin, until $\sum_{\forall f_{i,k} \in F_i} n_{i,k} = n_i^{\mathrm{R}}$. Let us consider an example with three flows $F_i = \{f_{i,1}, f_{i,2}, f_{i,3}\}$, where $p_{i,1} < p_{i,2} < p_{i,3}$. Assume that $n_i^{\mathrm{R}} = 8$, $m_{i,1} = 11$, $m_{i,2} = 6$, and $m_{i,3} = 2$. By Eq. (21), we have $n_{i,1} = \lceil \frac{11}{19} \cdot 8 \rceil = 5$, $n_{i,2} = \lceil \frac{6}{19} \cdot 8 \rceil = 3$, and $n_{i,3} = \lceil \frac{2}{19} \cdot 8 \rceil = 1$. Since $5 + 3 + 1 > n_i^{\mathrm{R}}$, we ask $f_{i,1}$ to give up one PRB. Therefore, $f_{i,1}$, $f_{i,2}$, $f_{i,3}$ will be allocated with 4, 3, and 1 PRBs, respectively. Lemma 2 presents the computation complexity of the scheduling module.

Lemma 2. *Given $\tilde{m}_{\mathrm{F}}$ flows and $\tilde{m}_{\mathrm{P}}$ PRBs, the scheduling module has time complexity of $O(\tilde{m}_{\mathrm{F}}(T + \lg \tilde{m}_{\mathrm{F}}) + \tilde{m}_{\mathrm{U}}\tilde{m}_{\mathrm{P}})$.*

Proof: Since the two-layer scheduling strategy dominates computation in this module, the worst case occurs when all flows are GBR and each eNB overloads (i.e., $L_j > 1$). For the UE-based scheduler, [27] shows that Eq. (15) spends T time to find $\varepsilon_{i,k}(t)$, so FLS takes $O(\tilde{m}_{\mathrm{F}}T)$ time. Finding $V_{i,t}$ for each UE by Eq. (16) spends $O(\tilde{m}_{\mathrm{F}})$ time. It also takes $O(\tilde{m}_{\mathrm{U}})$ time to find each UE's weight by Eq. (18). Then, the class-based MT method takes $O(\tilde{m}_{\mathrm{U}}\tilde{m}_{\mathrm{P}})$ time, as we have to check each (UE, PRB) pair in Eq. (17). So, the UE-based scheduler spends time of $O(\tilde{m}_{\mathrm{F}}T) + O(\tilde{m}_{\mathrm{F}}) + O(\tilde{m}_{\mathrm{U}}) + O(\tilde{m}_{\mathrm{U}}\tilde{m}_{\mathrm{P}}) = O(\tilde{m}_{\mathrm{F}}T) + O(\tilde{m}_{\mathrm{U}}\tilde{m}_{\mathrm{P}})$. For the flow-based scheduler, it takes $O(\tilde{m}_{\mathrm{F}})$ time to decide priorities of flows by Eq. (20). Also, using Eq. (21) to compute $n_{i,k}$ of each flow takes $O(\tilde{m}_{\mathrm{F}})$ time. When $\sum_{\forall f_{i,k} \in F_i} n_{i,k} > n_i^{\mathrm{R}}$, we sort flows by their priorities, which spends $O(\tilde{m}_{\mathrm{F}} \lg \tilde{m}_{\mathrm{F}})$ time, and ask flows to give up some PRBs, which takes $O(\tilde{m}_{\mathrm{F}})$ time. So, the flow-based scheduler spends time of $O(\tilde{m}_{\mathrm{F}}) + O(\tilde{m}_{\mathrm{F}}) + O(\tilde{m}_{\mathrm{F}} \lg \tilde{m}_{\mathrm{F}}) + O(\tilde{m}_{\mathrm{F}}) = O(\tilde{m}_{\mathrm{F}} \lg \tilde{m}_{\mathrm{F}})$. Thus, the scheduling module has time complexity of $O(\tilde{m}_{\mathrm{F}}T) + O(\tilde{m}_{\mathrm{U}}\tilde{m}_{\mathrm{P}}) + O(\tilde{m}_{\mathrm{F}} \lg \tilde{m}_{\mathrm{F}}) = O(\tilde{m}_{\mathrm{F}}(T + \lg \tilde{m}_{\mathrm{F}}) + \tilde{m}_{\mathrm{U}}\tilde{m}_{\mathrm{P}})$. □

5.3 Design of the Billing Module

The supply and demand theory in [7] shows that the charge $\hat{M}_i$ of a service affects the amount of demand $\hat{D}_i$ by a user to varying degrees. Specifically, given a scaling factor η, we can estimate a user's demand for the service by

$$\hat{D}_i = \eta \hat{M}_i^{-p_e}, \tag{22}$$

where p_e is a coefficient of *price elasticity*. To find p_e, we can adjust the service's charge to $\hat{M}_{i'}$ and measure the change of user's demand (from $\hat{D}_i$ to $\hat{D}_{i'}$) as follows:

$$\frac{\hat{D}_{i'}}{\hat{D}_i} = \frac{\eta \hat{M}_{i'}^{-p_e}}{\eta \hat{M}_i^{-p_e}} = \left(\frac{M_i}{M_{i'}}\right)^{p_e} \Rightarrow p_e = \frac{\ln(\hat{D}_{i'}/\hat{D}_i)}{\ln(\hat{M}_i/\hat{M}_{i'})}. \tag{23}$$

A larger p_e value (i.e., $\hat{D}_{i'}/\hat{D}_i$ increases or $\hat{M}_i/\hat{M}_{i'}$ decreases) implies that the decrease of service charge will increase user demand, so the service is more *elastic*. The work [31] points out that an applicable range of p_e for 3G and later systems is between 1.3 and 1.7, and it proposes the NLP pricing policy by Eq. (1), which computes the service's charge based on the network load, QoS levels, and user classes.

However, NLP is not completely fit for our EPS framework. We need to further tailor Eq. (1) by addressing three issues:

1) How to determine the network load L in a HetNet?
2) How to define the QoS level x in Eq. (1)?
3) How to apply peak and off-peak rates to NLP?

For the first issue, NLP simply defines L by the ratio of the number of PRBs spent by UEs in a macrocell to the number of PRBs offered by the eNB. Since EPS considers the HetNet scenario, we should compute the network load by

$$L = \frac{1}{|\mathcal{E}|} \sum\nolimits_{e_j \in \mathcal{E}} \min\{L_j, 1\}, \tag{24}$$

where $\mathcal{E}$ includes one macrocell eNB and all picocell eNBs in the macrocell. Based on Eq. (9), it is possible that $L_j > 1$ (i.e., the eNB overloads), so we use the term $\min\{L_j, 1\}$ in Eq. (24). We remark that since the coverage of a picocell is small, it is not suitable to compute the load L in each single cell, as UEs may easily handover to other cells due to mobility. Moreover, the eNB controlling module in Section 5.1 works based on one macrocell along with its covered picocells. Thus, it is natural to take the average load of eNBs in $\mathcal{E}$ as the network load by Eq. (24).

For the second issue, we refer to QCI in Table 3 to define the QoS level. In particular, given the *QCI priority* y of a flow, we calculate its QoS level by the following function:

$$f(y) = 10 - \lceil y \rceil. \tag{25}$$

From Table 3, since $0.5 \leq y \leq 9$, the output of $f(y)$ must be a positive integer and its value is between 1 and 9. In this way, we can make sure that the difference between the QoS levels of any two flows will not be too large (in particular, no more than 8). Thus, we can avoid the situation where a user will be charged too much (or less) when he/she uses a certain flow. Moreover, since GBR flows usually have larger QCI priorities and the eNB may spend more resource to meet their QoS requirements (e.g., video flows), we can ask users to pay more money to increase operator profit by Eqs. (1) and (25) when they use large-demand GBR applications. Note that the previous PARS framework [9] directly uses the QCI "index" to be the QoS level in Eq. (5). Since PARS also limits the QoS level between 1 and 9, it simply sets the QoS level to 9 when a flow has a QCI index larger than 9 (e.g., 65, 66, 69, 70, 75, and 79 in Table 3). In this case, PARS cannot differentiate these flows by giving them different QoS levels.

For the third issue, we use two thresholds δ_{peak} and δ_{off} on L to check if the network load is heavy or light, respectively. Then, we adjust the fee charged to users accordingly. When $L \geq \delta_{\mathrm{peak}}$, we add an *additional fee* C_{peak} to the charge to reflect the peak rate, so as to increase operator profit and alleviate network congestion. On the other hand, when $L \leq \delta_{\mathrm{off}}$, we deduct a *bonus fee* C_{off} from the charge to reflect the off-peak rate, so as to increase resource utilization.

Based on the above designs, we amend Eq. (1) in the billing module. Suppose that a user has class of ξ_i. He/She uses a flow $f_{i,k}$ with QCI priority $y_{i,k}$ that consumes $n_{i,k}$ PRBs. Then, we adopt three cases to compute the fee charged to the user:

- Case of $L \leq \delta_{\mathrm{off}}$ (i.e., off-peak rate):

$$M_{i,k} = [C_{\mathrm{V}}(\xi_i) - C_{\mathrm{off}}] \times [\hat{e} - \hat{e}^{-f(y_{i,k})}] L n_{i,k}. \tag{26}$$

- Case of $\delta_{\mathrm{off}} < L < \delta_{\mathrm{peak}}$ (i.e., normal rate):

$$M_{i,k} = C_{\mathrm{V}}(\xi_i) \times [\hat{e} - \hat{e}^{-f(y_{i,k})}] L n_{i,k}. \tag{27}$$

TABLE 3: QCI table defined by the LTE-A standard (Release 15) [12].

index	flow	priority	example services
1	GBR	2	conversational voice (e.g., VoIP)
2	GBR	4	conversational video (live streaming)
3	GBR	3	real-time gaming, V2X messages
4	GBR	5	video (buffered streaming)
65	GBR	0.7	MCPTT voice
66	GBR	2	non-MCPTT voice
75	GBR	2.5	V2X messages
5	non-GBR	1	IMS signalling
6	non-GBR	6	video (buffered streaming), TCP-based
7	non-GBR	7	voice, video (live streaming)
8	non-GBR	8	video (buffered streaming), TCP-based
9	non-GBR	9	video (buffered streaming), TCP-based
69	non-GBR	0.5	mission critical delay sensitive signalling
70	non-GBR	5.5	mission critical data
79	non-GBR	6.5	V2X messages
[Note]	V2X: vehicle to everything, MCPTT: mission critical push to talk IMS: IP multimedia subsystem, TCP-based: www, email, chat, ftp, p2p, and so on		

- Case of $L \geq \delta_{\text{peak}}$ (i.e., peak rate):

$$M_{i,k} = [C_{\text{V}}(\xi_i) + C_{\text{peak}}] \times [\hat{e} - \hat{e}^{-f(y_{i,k})}]Ln_{i,k}. \quad (28)$$

We discuss the meanings of our pricing functions. All functions have the same term of $[\hat{e} - \hat{e}^{-f(y_{i,k})}]Ln_{i,k}$, so the fee is proportional to the network load L and the number of PRBs $n_{i,k}$ spent by a flow. Thus, when L increases or the user spends more PRBs, the fee will be raised accordingly, and vice versa. Here, the term $[\hat{e} - \hat{e}^{-f(y_{i,k})}]$ reflects the price elasticity p_e defined in Eq. (23), where each flow has its elasticity based on the QoS level $f(y_{i,k})$. When the flow has a larger $f(y_{i,k})$ value (i.e., a higher QCI priority), the value of $[\hat{e} - \hat{e}^{-f(y_{i,k})}]$ increases. It means that the flow is less elastic, and we can raise the fee as the price has a smaller effect on the user's demand for the flow. In other words, the user would not greatly reduce the flow's demand if we raise the fee. Observing from Eqs. (26)–(28), each user is charged with a unit fee $C_{\text{V}}(\xi_i)$ based on the class ξ_i. We have $C_{\text{V}}(\xi_{\text{G}}) > C_{\text{V}}(\xi_{\text{S}}) > C_{\text{V}}(\xi_{\text{B}})$, so gold, silver, and bronze users will pay high, medium, and low rates, respectively. To encourage users boosting their demands as the network utilization is low, we reduce the unit fee to $[C_{\text{V}}(\xi_i) - C_{\text{off}}]$ in Eq. (26). When the network is saturated, we raise the unit fee to $[C_{\text{V}}(\xi_i) + C_{\text{peak}}]$ in Eq. (28) to avoid users consuming too much resource, so as to mitigate congestion. Note that the amount of money that a user should pay is the sum of fees $M_{i,k}$ of all flows $f_{i,k}$ owned by the user. Lemma 3 gives the time complexity of the billing module.

Lemma 3. *The billing module spends time of $O(\tilde{m}_{\text{E}} + \tilde{m}_{\text{F}})$.*

Proof: We first compute the load L by Eq. (24), which takes $O(\tilde{m}_{\text{E}})$ time. Then, each flow is charged based on Eqs. (26), (27), or (28), which spends $O(\tilde{m}_{\text{F}})$ time. Thus, the billing module requires time of $O(\tilde{m}_{\text{E}} + \tilde{m}_{\text{F}})$. □

5.4 Discussion

We discuss the rationale of our EPS framework. As mentioned in Section 3, many studies view resource scheduling, pricing, and energy saving as independent issues. However, they do affect each other in LTE-A HetNets. In particular, we can save energy of eNBs by DTX on the premise that UEs are properly assigned to eNBs for resource scheduling. Moreover, the supply and demand theory shows the correlation between user demands (for resource scheduling) and pricing.

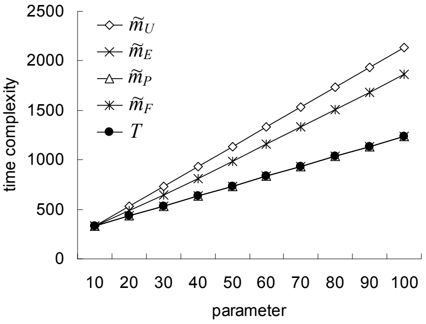


Fig. 2: Effects of different parameters on EPS's time complexity.

With this motivation, EPS uses three modules to co-address these issues, as shown in Fig. 1. The eNB controlling module organizes picocell eNBs into groups, and allows each CR to arrange UEs in its group with two objectives. One is to alleviate congestion in some picocells to make sure that an eNB never has an excessive load, so we can facilitate the scheduling module accordingly. The other objective is to make the picocell eNBs serving just few UEs become idle, so as to let them sleep to save energy.

The principle of the scheduling module is to give a high priority to GBR flows for QoS consideration. To cooperate with the billing module and improve network throughput, it finds the transmission amount of GBR data by FLS to meet their QoS demands, and adopts the class-based MT method to let UEs with higher classes or better channel quality to get PRBs first. Then, the DAPA strategy gives precedence to urgent flows to reduce packet dropping, and allots PRBs to flows based on their M-LWDF priorities derived by QCIs.

The billing module considers the supply and demand theory. It tailors the NLP policy by computing the load of a HetNet, finding QoS levels of flows, and applying both peak and off-peak rates. In this way, we can increase operator profit and mitigate network congestion by raising the charge to each user as the network is saturated. Moreover, when the HetNet's load becomes light, the billing module encourages users to increase their service utilization by reducing the charge. Theorem 1 analyzes EPS's computation complexity.

Theorem 1. *The time complexity of the EPS framework is $O(\tilde{m}_{\text{U}}(\tilde{m}_{\text{E}} + \tilde{m}_{\text{P}}) + \tilde{m}_{\text{F}}(T + \lg \tilde{m}_{\text{F}}))$.*

Proof: Based on Lemmas 1, 2, and 3, the time complexity of EPS is $O(\tilde{m}_{\text{U}}\tilde{m}_{\text{E}}) + O(\tilde{m}_{\text{F}}(T + \lg \tilde{m}_{\text{F}}) + \tilde{m}_{\text{U}}\tilde{m}_{\text{P}}) + O(\tilde{m}_{\text{E}} + \tilde{m}_{\text{F}}) = O(\tilde{m}_{\text{U}}(\tilde{m}_{\text{E}} + \tilde{m}_{\text{P}}) + \tilde{m}_{\text{F}}(T + \lg \tilde{m}_{\text{F}}))$. □

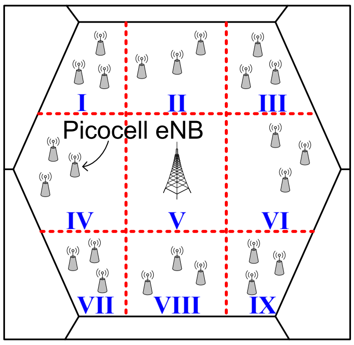


Fig. 3: Network topology in our simulations.

Theorem 1 indicates that the computation time of EPS (in the worst case) depends on the number of UEs $\tilde{m}_{\rm U}$, the number of eNBs $\tilde{m}_{\rm E}$, the number of PRBs $\tilde{m}_{\rm P}$, the number of flows $\tilde{m}_{\rm F}$, and period length T. Fig. 2 shows their effects on EPS's time complexity. In the experiment, when we vary the value of a parameter from 10 to 100, the values of all other parameters are fixed to 10. We can observe that $\tilde{m}_{\rm U}$ has the greatest effect, followed by $\tilde{m}_{\rm F}$. Then, $\tilde{m}_{\rm E}$, $\tilde{m}_{\rm P}$, and T have the same effect.

We remark that operators usually offer a number of pricing options, where each option is associated with a basic rate and a limit on data usage. In general, a higher basic rate, a larger limit on data usage. Thus, these options actually correspond to different classes of users. When a user consumes resource more than the limit, the user needs to pay extra charge for the amount of resource that is over the limit. However, using a static pricing policy cannot deal with this situation. That is why we choose to develop a dynamic pricing policy in EPS. In particular, the operator can refer to our billing module to calculate the amount of extra fee charged to a user when he/she spends resource more than the limit, so as to increase both profit and system utilization.

6 Simulation Study

We evaluate EPS's performance by MATLAB. Fig. 3 gives the topology of eNBs, where the macrocell is cut into nine regions. Except for region V, there are three picocell eNBs deployed in each region. The macrocell eNB has cell range of 1500m and transmitted power of 46dBm. Its channel bandwidth is 20MHz, which offers 100 PRBs/TTI. Each picocell eNB has cell range of 250m and transmitted power of 30dBm. Its channel bandwidth is 5MHz, which offers 25 PRBs/TTI. A picocell eNB can save 50% and 90% amount of energy when it keeps in the sniff and sleep modes, respectively [55], [56]. Besides, we set $\delta^{\rm L}_{\rm load} = 30\%$ and $\delta^{\rm H}_{\rm load} = 125\%$.

There are 60, 180, 300, 420, 540, 660, 780, 900, 1020, and 1140 UEs in the HetNet, and two distributions of UEs are modeled. In the *uniform distribution*, UEs are evenly scattered over all regions in Fig. 3, so each picocell could have a similar number of UEs. In the *hotspot distribution*, around $4/9$ of UEs locate in regions III and VII, and the residual UEs are uniformly distributed over other regions. The ratio of gold, silver, and bronze UEs is 1:1:1. In addition, we consider three types of traffic flows: 1) 8.4kbps VoIP flow with QCI = 1, 2) 242kbps H.264 video flow with QCI = 2, and 3) 12kbps constant-bit-rate flow with QCI = 6. Each UE produces one or two flows. The user class and flows of a UE will not change in the simulations. However, we apply the supply and demand theorem to adjust the traffic demand of each flow, where the scaling factor η in Eq. (22) is set to 2×10^5 [7].

Based on the specification of LTE-A [52], we model the environmental noise by a Gaussian white noise whose power spectral density is -174dBm/Hz. The path-loss effect from eNB e_j to UE u_i is estimated by a log-distance model: $128.1 + 37.6 \log Z(u_i, e_j)$ for a macrocell and $38 + 30 \log(10^3 Z(u_i, e_j))$ for a picocell, where $Z(u_i, e_j)$ is their distance in kilometers. Besides, LTE-A adopts a zero-mean log-normal distribution to measure the effect of shadowing fading. Its standard deviation is set to 10dB and 6dB for a macrocell or picocell, respectively.

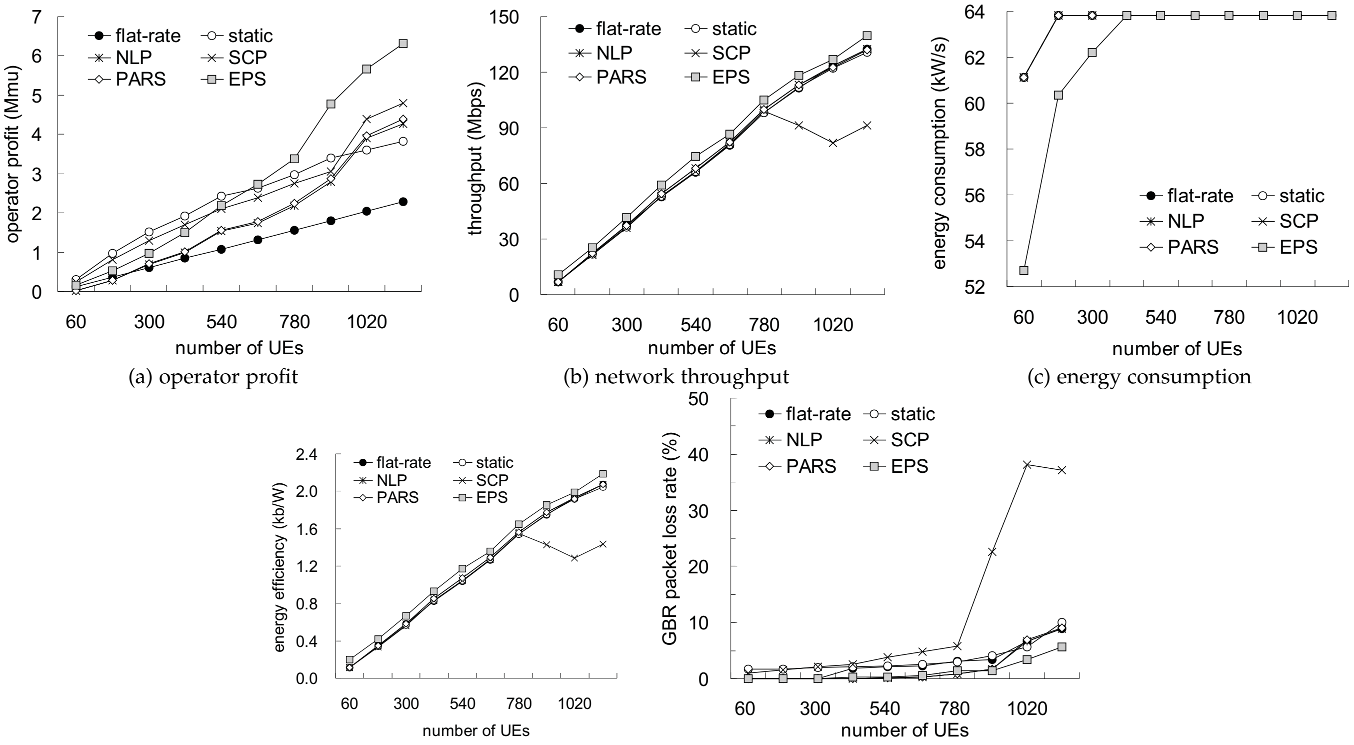
We compare EPS with the flat-rate, static pricing, NLP, SCP, and PARS methods discussed in Section 3.2. According to [9], [32], [57], we set their parameters as follows: For the flat-rate method, each user is charged with 2000mus. For the static pricing method, we set $C_{\rm P}(\xi_{\rm G}) = 11$, $C_{\rm P}(\xi_{\rm S}) = 6$, and $C_{\rm P}(\xi_{\rm B}) = 4$. In NLP, we set $C_{\rm V}(\xi_{\rm G}) = 0.9$, $C_{\rm V}(\xi_{\rm S}) = 0.7$, $C_{\rm V}(\xi_{\rm B}) = 0.5$, $C_{\rm F} = 2.6$, and $\alpha = 1$. In SCP, we set $C_{\rm P}(\xi_{\rm G}) = 9$, $C_{\rm P}(\xi_{\rm S}) = 8$, $C_{\rm P}(\xi_{\rm B}) = 4$, and $\beta = 520$. In PARS, we set $C_{\rm V}(\xi_{\rm G}) = 0.9$, $C_{\rm V}(\xi_{\rm S}) = 0.7$, $C_{\rm V}(\xi_{\rm B}) = 0.5$, and $C_{\rm F} = 2.6$. In EPS, we set $C_{\rm V}(\xi_{\rm G}) = 0.9$, $C_{\rm V}(\xi_{\rm S}) = 0.7$, $C_{\rm V}(\xi_{\rm B}) = 0.5$, $C_{\rm F} = 2.6$, and $C_{\rm off} = C_{\rm peak} = 0.1$. Except for the flat-rate method, the unit of price is mu/PRB, where "mu" is the abbreviation of *monetary unit*. For the flat-rate, static pricing, NLP, and SCP methods, we use the resource scheduling scheme in PARS to allocate PRBs. Besides, we apply the DTX technique to these methods. Specifically, a picocell eNB can be turned off if it has no UE to serve.

We measure the amount of operator profit, network throughput, energy consumption, energy efficiency, and packet loss by different methods. Network throughput is defined by the number of data bits received by UEs in each second (measured in megabits per second, i.e., Mbps). Energy efficiency is defined by the ratio of network throughput to energy consumption of eNBs (measured in kilobits per watt, i.e., kb/W).

6.1 Uniform Distribution of UEs

We first evaluate performance in the uniform distribution of UEs. Fig. 4(a) gives the amount of operator profit in 100 seconds, where 1Mmu = 10^6mus. Obviously, when the number of UEs grows, the operator can increase its profit. The amount of profit increase is linear in the flat-rate method, as it simply keeps the fee regardless of the network load. Comparing with the static pricing method, NLP, SCP, and PARS allow the operator getting more profit under a heavy load (i.e., > 900 UEs). On the other hand, EPS adopts off-peak rates to attract users when the load is light (i.e., < 540 UEs), so its profit is slightly below the static and SCP methods. However, when there are more than 780 UEs, EPS changes to use peak rates, so it helps the operator get the most profit under a heavy load. The above behavior shows the flexibility of EPS's pricing policy.

Fig. 4(b) shows network throughput. As the flat-rate, static pricing, NLP, and SCP methods adopt the scheduling scheme of PARS, they have similar throughput. However, SCP gives low-class users more penalty by Eqs. (3) and (4) under a heavy load (i.e., > 780 UEs), which discourages them using more resource (caused by the effect of supply and demand theory). Thus, SCP's throughput drops when the number of UEs exceeds 780. That is why the curve of SCP is non-monotonic. Our EPS framework always has the highest throughput due



(a) operator profit

(b) network throughput

(c) energy consumption

(d) energy efficiency

(e) packet loss

Fig. 4: Comparison on performance in the uniform distribution of UEs.

to two reasons. First, it reduces the charge to encourage users increasing traffic demands as the network load is light. Second, the eNB controlling module arranges UEs to balance loads of picocells and thus alleviates network congestion.

Fig. 4(c) gives the amount of energy consumed by eNBs per second. Though the flat-rate, static pricing, NLP, SCP, and PARS methods use the DTX technique, they fail to save energy of eNBs. In particular, only when the network load is very light (i.e., with only 60 UEs) can these methods find out idle eNBs to turn them off. On the contrary, EPS allows CRs actively arranging UEs to make the picocell eNBs serving just few UEs become idle. Thus, it reduces much more energy spent by eNBs with 60 UEs. Besides, EPS can still save energy of eNBs when there are no more than 300 UEs.

We then study energy efficiency of different methods, as presented in Fig. 4(d). Specifically, higher energy efficiency means that eNBs can better utilize their transmitted power to send more data. From Fig. 4(c), since each method makes eNBs consume the same amount of energy when there are more than 300 UEs, the trend of energy efficiency in Fig. 4(d) will be similar to that of network throughput in Fig. 4(b). Therefore, the curve of SCP is also non-monotonic. From Fig. 4(d), we observe that our EPS framework always keeps the highest energy efficiency, which verifies that it can increase network throughput while save energy of eNBs.

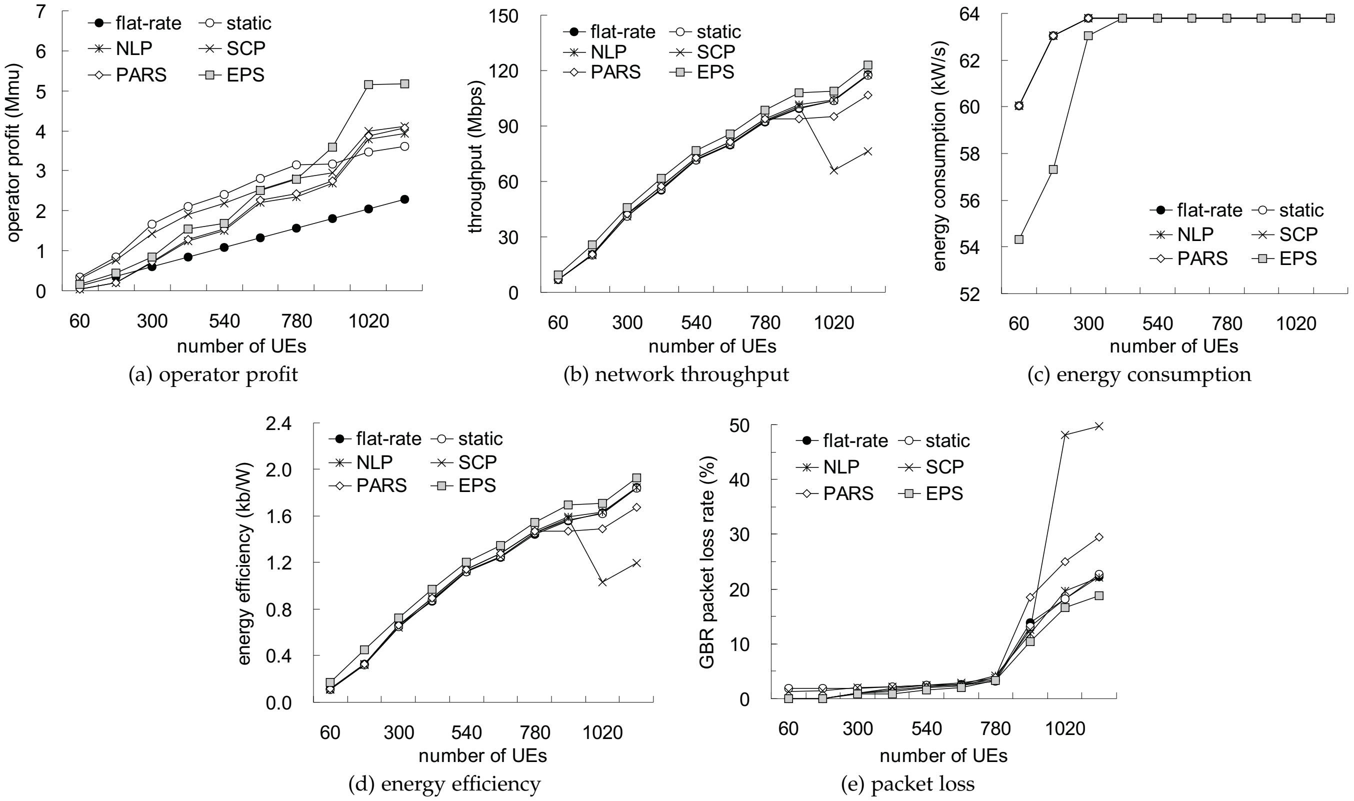
Fig. 4(e) shows the packet loss rate of GBR flows due to expiration. When the network load becomes heavy (i.e., > 780 UEs), the loss rate of each method starts growing. However, the loss rate of SCP drastically increases in this case. The reason is that SCP forces users with low classes to pay much more money, which frightens them out of using more GBR services (and leave many GBR packets to be discarded). On the other hand, since our scheduling module lets urgent flows acquire PRBs first, the EPS framework can result in a lower loss rate as comparing with other methods.

6.2 Hotspot Distribution of UEs

We then investigate performance in the hotspot distribution of UEs. Since many UEs reside in regions III and VII of the macrocell, they will cause network congestion in these regions and get fewer PRBs than that in the uniform distribution. Thus, except for the flat-rate method, the amount of operator profit decreases in all other methods in Fig. 5(a) as comparing with Fig. 4(a). However, our EPS framework still keeps its pricing flexibility and thus helps the operator get the most profit when the network load becomes heavy in this distribution of UEs.

Fig. 5(b) gives the amount of network throughput. Comparing with Fig. 4(b), all methods result in lower throughput due to network congestion in regions III and VII. SCP still encounters significant throughput loss as the number of UEs is above 900 due to the excessive charge to low-class users by Eqs. (3) and (4), which makes its curve non-monotonic. Thanks to the design of three modules, our EPS framework keeps the highest throughput among all methods, which demonstrates its throughput effectiveness in LTE-A HetNets.

Fig. 5(c) measures the amount of energy spent by eNBs. Interestingly, when the number of UEs is 180, each method can save more energy as comparing with that in Fig. 4(c). The reason is that only $5/9$ of UEs are randomly scattered over the six regions (except for III and VII). In this case, there is higher possibility that we can find out more idle eNBs to turn them off and save energy accordingly. Then, Fig. 5(d) shows the result of energy efficiency. Again, the curve of each method in Fig. 5(d) will be similar to those in Fig. 5(b), because each method results in the same amount of energy consumption of eNBs when there are more than 300 UEs (referring to Fig. 5(c)). That is why the curve of SCP is also non-monotonic in Fig. 5(d). From Figs. 5(c) and (d), we show that EPS can save more



(a) operator profit

(b) network throughput

(c) energy consumption

(d) energy efficiency

(e) packet loss

Fig. 5: Comparison on performance in the hotspot distribution of UEs.

energy of eNBs and result in the highest energy efficiency in the hotspot distribution of UEs.

Fig. 5(e) compares the GBR packet loss rates of different methods. Since $4/9$ of UEs congregate in regions III and VII, these UEs cause network congestion in their cells. In this case, some picocell eNBs may not have enough PRBs to serve their GBR flows. Such a situation becomes worse as there are more UEs in the network. That is why the loss rate significantly increases in each method when the number of UEs is above 780. Since our EPS framework arranges UEs to balance cell loads and also serves urgent flows first, it can result in the lowest loss rate as comparing with other methods.

6.3 Daily Traffic Profile in Europe

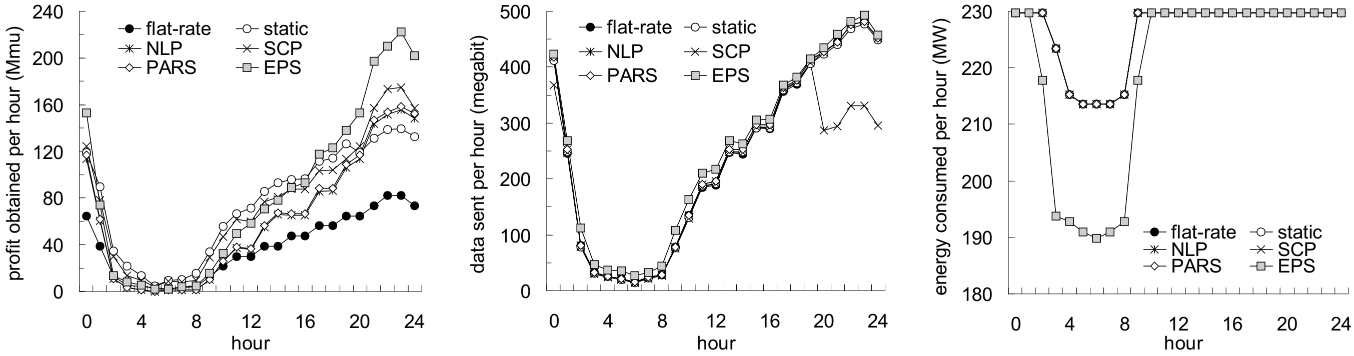
As the experiments in Sections 6.1 and 6.2 have simulation time of only 100 seconds, we refer to the daily traffic profile in Europe [58] to imitate user demands in one day, where the maximum number of UEs is 1140. Fig. 6(a) presents the amount of operator profit in each hour. We observe that the off-peak time is from the 2nd to 9th hours and the peak time is from the 17th to 24th hours. Even in the peak time, the flat-rate method cannot greatly increase operator profit, which shows the necessity of using a dynamic pricing policy. On the other hand, the operator can obtain the most profit during the peak time in EPS, which verifies its flexibility in pricing.

Fig. 6(b) shows the amount of data transmission per hour. As discussed earlier, SCP incurs serious throughput loss in the peak time, since it asks users to pay more money but does not provide better services. On the contrary, although EPS charges users with higher fees in the peak time, it adaptively distributes UEs over picocells and offers better service quality. That is why EPS does not encounter throughput loss in the peak time. Moreover, EPS deducts C_{off} from the fee by Eq. (26) in the off-peak time, so it can encourage users increasing their demands when the network load becomes light.

Fig. 6(c) gives the amount of energy consumed by eNBs in each hour. Since more eNBs can become idle in the off-peak time (especially from the 3rd to 8th hours), each method can turn off unused eNBs and save energy accordingly. Thanks to the arrangement of UEs by the eNB controlling module, our EPS framework greatly reduces energy consumption of eNBs in the off-peak time. This experiment shows that EPS can better support green communications than others.

7 Conclusion

In LTE-A HetNets, the issues of resource scheduling, pricing, and energy saving have great impact on performance. This paper thus develops the EPS framework with three modules to co-address these issues. The eNB controlling module groups picocells and asks the CR to manage UEs in its group, so as to balance loads of picocell eNBs and make some eNBs sleep to save energy. The scheduling module adopts a two-layer scheduling strategy to serve GBR flows first to meet their QoS demands, and allots PRBs to other flows via the class-based MT method. The billing module considers the effect of supply and demand theory and introduces both peak and off-peaks rates into the pricing policy. Through simulations by MATLAB, we show that EPS has better performance on operator profit, network throughput, energy consumption, and packet loss, as compared with the flat-rate, static pricing, NLP, SCP, and PARS methods. For the future work, we will investigate more sophisticated clustering schemes of picocell eNBs (e.g., varying δ_g discussed in Section 4 based on various conditions), and evaluate their effects on performance.



(a) operator profit (b) data transmission (c) energy consumption

Fig. 6: Comparison on performance by the daily traffic profile in Europe.

References

[1] Cisco, “Visual networking index: forecast and methodology, 2016–2021,” 2017. [Online]. Available: http://www.cisco.com/

[2] F. Capozzi, G. Piro, L.A. Grieco, G. Boggia, and P. Camarda, “Downlink packet scheduling in LTE cellular networks: key design issues and a survey,” *IEEE Comm. Surveys & Tutorials*, vol. 15, no. 2, pp. 678–700, 2013.

[3] S. Singh and J.G. Andrews, “Joint resource partitioning and offloading in heterogeneous cellular networks,” *IEEE Trans. Wireless Comm.*, vol. 13, no. 2, pp. 888–901, 2014.

[4] Y.C. Wang and C.A. Chuang, “Efficient eNB deployment strategy for heterogeneous cells in 4G LTE systems,” *Computer Networks*, vol. 79, pp. 297–312, 2015.

[5] A. Fehske, G. Fettweis, J. Malmodin, and G. Biczok, “The global footprint of mobile communications: the ecological and economic perspective,” *IEEE Comm. Magazine*, vol. 49, no. 8, pp. 55–62, 2011.

[6] C.A. Gizelis and D.D. Vergados, “A survey of pricing schemes in wireless networks,” *IEEE Comm. Surveys & Tutorials*, vol. 13, no. 1, pp. 126–145, 2011.

[7] S. Lanning, D. Mitra, Q. Wang, and M. Wright, “Optimal planning for optical transport networks,” *Philosophical Trans. the Royal Society of London A*, vol. 358, no. 1773, pp. 2183–2196, 2000.

[8] K.J. Zou, K.W. Yang, M. Wang, B. Ren, J. Hu, J. Zhang, M. Hua, and X. You, “Network synchronization for dense small cell networks,” *IEEE Wireless Comm.*, vol. 22, no. 2, pp. 108–117, 2015.

[9] Y.C. Wang and T.Y. Tsai, “A pricing-aware resource scheduling framework for LTE networks,” *IEEE/ACM Trans. Networking*, vol. 25, no. 3, pp. 1445–1458, 2017.

[10] C. Cox, “Quality of service, policy and charging,” in *An Introduction to LTE: LTE, LTE-Advanced, SAE, VoLTE and 4G Mobile Communications*, New York: Wiley, 2014.

[11] D. Astely, E. Dahlman, A. Furuskar, Y. Jading, M. Lindstrom, and S. Parkvall, “LTE: the evolution of mobile broadband,” *IEEE Comm. Magazine*, vol. 47, no. 4, pp. 44–51, 2009.

[12] ETSI, “Policy and charging control architecture (release 15),” 3GPP TS 23.203 V15.0.0, 2017.

[13] Y.C. Wang, S.R. Ye, and Y.C. Tseng, “A fair scheduling algorithm with traffic classification in wireless networks,” *Computer Comm.*, vol. 28, no. 10, pp. 1225–1239, 2005.

[14] S. Schwarz, C. Mehlfuhrer, and M. Rupp, “Throughput maximizing multiuser scheduling with adjustable fairness,” *Proc. IEEE Int'l Conf. Comm.*, 2011, pp. 1–5.

[15] S. Ali and M. Zeeshan, “A utility based resource allocation scheme with delay scheduler for LTE service-class support,” *Proc. IEEE Wireless Comm. and Networking Conf.*, 2012, pp. 1450–1455.

[16] W.K. Lai and C.L. Tang, “QoS-aware downlink packet scheduling for LTE networks,” *Computer Networks*, vol. 57, no. 7, pp. 1689–1698, 2013.

[17] M.B. Shahab, M.A. Wahla, and M.T. Mushtaq, “Downlink resource scheduling technique for maximized throughput with improved fairness and reduced BLER in LTE,” *Proc. IEEE Int'l Conf. Telecomm. and Signal Processing*, 2015, pp. 163–167.

[18] M. Iturralde, T.A. Yahiya, A. Wei, and A.L. Beylot, “Resource allocation using Shapley value in LTE networks,” *Proc. IEEE Int'l Symp. Personal Indoor and Mobile Radio Comm.*, 2011, pp. 31–35.

[19] F. Huang, V. Veque, and J. Tomasik, “A Pareto-optimal approach for resource allocation on the LTE downlink,” *Proc. IEEE Int'l Conf. Comm.*, 2016, pp. 1–7.

[20] Y.C. Wang, “A two-phase dispatch heuristic to schedule the movement of multi-attribute mobile sensors in a hybrid wireless sensor network,” *IEEE Trans. Mobile Computing*, vol. 13, no. 4, pp. 709–722, 2014.

[21] Y.C. Wang and D.R. Jhong, “Efficient allocation of LTE downlink spectral resource to improve fairness and throughput,” *Int'l J. Comm. Systems*, vol. 30, no. 14, pp. 1–13, 2017.

[22] B. Liu, H. Tian, and L. Xu, “An efficient downlink packet scheduling algorithm for real time traffics in LTE systems,” *Proc. IEEE Consumer Comm. and Networking Conf.*, 2013, pp. 364–369.

[23] C. Wang and Y.C. Huang, “Delay-scheduler coupled throughput-fairness resource allocation algorithm in the long-term evolution wireless networks,” *IET Comm.*, vol. 8, no. 17, pp. 3105–3112, 2014.

[24] Y.C. Wang and S.Y. Hsieh, “Service-differentiated downlink flow scheduling to support QoS in long term evolution,” *Computer Networks*, vol. 94, pp. 344–359, 2016.

[25] D. Samia, B. Ridha, and A. Wei, “Resource allocation using Nucleolus value in downlink LTE networks,” *Proc. IEEE Symp. Computers and Comm.*, 2016, pp. 250–254.

[26] H. Luo, S. Ci, D. Wu, J. Wu, and H. Tang, “Quality-driven cross-layer optimized video delivery over LTE,” *IEEE Comm. Magazine*, vol. 48, no. 2, pp. 102–109, 2010.

[27] G. Piro, L.A. Grieco, G. Boggia, R. Fortuna, and P. Camarda, “Two-level downlink scheduling for real-time multimedia services in LTE networks,” *IEEE Trans. Multimedia*, vol. 13, no. 5, pp. 1052–1065, 2011.

[28] Q. Liu and C.W. Chen, “Smart downlink scheduling for multimedia streaming over LTE networks with hard handoff,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 25, no. 11, pp. 1815–1829, 2015.

[29] J. Cushnie, D. Hutchison, and H. Oliver, “Evolution of charging and billing models for GSM and future mobile Internet services,” in *Quality of Future Internet Services*, New York: Springer, 2002, pp. 312–323.

[30] P. Marbach, “Analysis of a static pricing scheme for priority services,” *IEEE/ACM Trans. Networking*, vol. 12, no. 2, pp. 312–325, 2004.

[31] E. Wallenius and T. Hamalainen, “Pricing model for 3G/4G networks,” *Proc. IEEE Int'l Symp. Personal, Indoor and Mobile Radio Comm.*, 2002, pp. 187–191.

[32] U. Mir and L. Nuaymi, “LTE pricing strategies,” *Proc. IEEE Vehicular Technology Conf.*, 2013, pp. 1–6.

[33] H. Shajaiah, A. Abdelhadi, and C. Clancy, “A price selective centralized algorithm for resource allocation with carrier aggregation in LTE cellular networks,” *Proc. IEEE Wireless Comm. and Networking Conf.*, 2015, pp. 813–818.

[34] Ramneek, P. Hosein, and W. Seok, “Load metric for QoS-enabled cellular networks and its possible use in pricing strategies,” *Proc. IEEE Symp. Wireless Technology and Applications*, 2014, pp. 30–35.

[35] A. Abdelhadi and C. Clancy, “A robust optimal rate allocation algorithm and pricing policy for hybrid traffic in 4G-LTE,” *Proc. IEEE Int'l Symp. Personal, Indoor, and Mobile Radio Comm.*, 2013, pp. 2185–2190.

[36] Y. C. Wang and S. Lee, “Small-cell planning in LTE HetNet to improve energy efficiency,” *Int'l J. Comm. Systems*, vol. 31, no. 5, pp. 1–18, 2018.

[37] R. Imran, M. Shukair, N. Zorba, O. Kubbar, and C. Verikoukis, “A novel energy saving MIMO mechanism in LTE systems,” *Proc. IEEE Int'l Conf. Comm.*, 2013, pp. 2449–2453.

[38] S. Jin, X. Ma, and W. Yue, “Energy-saving strategy for green cognitive radio networks with an LTE-advanced structure,” *J. Comm. and Networks*, vol. 18, no. 4, pp. 610–618, 2016.

[39] Y.C. Wang and H.Y. Ko, “Energy-efficient downlink resource scheduling for LTE-A networks with carrier aggregation,” *J. Information Science and Engineering*, vol. 33, no. 1, pp. 123–141, 2017.

[40] P. Frenger, P. Moberg, J. Malmodin, Y. Jading, and I. Godor, "Reducing energy consumption in LTE with cell DTX," *Proc. IEEE Vehicular Technology Conf.*, 2011, pp. 1–5.

[41] K. Abdallah, I. Cerutti, and P. Castoldi, "Energy-efficient coordinated sleep of LTE cells," *Proc. IEEE Int'l Conf. Comm.*, 2012, pp. 5238–5242.

[42] N. Saxena, B.J.R. Sahu, and Y.S. Han, "Traffic-aware energy optimization in green LTE cellular systems," *IEEE Comm. Letters*, vol. 18, no. 1, pp. 38–41, 2014.

[43] Y.L. Chung, "An efficient power-saving transmission mechanism in LTE macrocell-femtocell hybrid networks," *Proc. Int'l Conf. Information Networking*, 2014, pp. 176–180.

[44] R. Combes, S.E. Elayoubi, A. Ali, L. Saker, and T. Chahed, "Optimal online control for sleep mode in green base stations," *Computer Networks*, vol. 78, no. 26, pp. 140–151, 2015.

[45] Y.C. Wang and S.J. Liu, "Minimum-cost deployment of adjustable readers to provide complete coverage of tags in RFID systems," *J. Systems and Software*, vol. 134, pp. 228–241, 2017.

[46] K.Y. Lin, J.Y. Chen, F.C. Ren, and C.J. Chang, "TAPS: traffic-aware power saving scheme for clustered small cell base stations in LTE-A," *Proc. IEEE Vehicular Technology Conf.*, 2015, pp. 1–5.

[47] T.Z. Oo, N.H. Tran, W. Saad, D. Niyato, Z. Han, and C.S. Hong, "Offloading in HetNet: a coordination of interference mitigation, user association, and resource allocation," *IEEE Trans. Mobile Computing*, vol. 16, no. 8, pp. 2276–2291, 2017.

[48] J. Xu, S. Wu, L. Xu, N. Zhang, and Q. Zhang, "Green-oriented user-satisfaction aware WiFi offloading in HetNets," *IET Comm.*, vol. 12, no. 5, pp. 501–508, 2018.

[49] M. Simsek, M. Bennis, and I. Guvenc, "Learning based frequency- and time-domain inter-cell interference coordination in HetNets," *IEEE Trans. Vehicular Technology*, vol. 64, no. 10, pp. 4589–4602, 2015.

[50] H. Zhou, Y. Ji, X. Wang, and S. Yamada, "eICIC configuration algorithm with service scalability in heterogeneous cellular networks," *IEEE/ACM Trans. Networking*, vol. 25, no. 1, pp. 520–535, 2017.

[51] C. Mehlfuhrer, M. Wrulich, J.C. Ikuno, D. Bosanska, and M. Rupp, "Simulating the long term evolution physical layer," *Proc. European Signal Processing Conf.*, 2009, pp. 1471–1478.

[52] ETSI, "Evolved universal terrestrial radio access (E-UTRA); physical layer procedures (release 14)," 3GPP TS 36.213 V14.1.0, 2016.

[53] Y.C. Wang and Y.C. Tseng, "Packet fair queuing algorithms for wireless networks," in *Design and Analysis of Wireless Networks*, Hauppauge: Nova Science Publishers, 2005.

[54] K.J. Astrom and B. Wittenmark, *Computer Controlled Systems: Theory and Design*, New York: Dover Publications, 2012.

[55] K. Hiltunen, "Utilizing eNodeB sleep mode to improve the energy-efficiency of dense LTE networks," *Proc. IEEE Int'l Symp. Personal, Indoor, and Mobile Radio Comm.*, 2013, pp. 3249–3253.

[56] E. Mugume and D.K.C. So, "Sleep mode mechanisms in dense small cell networks," *Proc. IEEE Int'l Conf. Comm.*, 2015, pp. 192–197.

[57] A. Belghith, S. Trabelsi, and B. Cousin, "Realistic per-category pricing schemes for LTE users," *Proc. IEEE Int'l Symp. Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 2014, pp. 429–435.

[58] G. Auer, O. Blume, and V. Giannini, "Energy efficiency analysis of the reference systems, areas of improvements and target breakdown," EARTH Project Report, Deliverable D2.3, 2012.