Abstract—Long term evolution-advanced (LTE-A) is a mobile communication standard widely used for broadband wireless access. It allows large macrocells and small picocells coexisting in a heterogeneous network (HetNet) to increase throughput and facilitate deployment. To handle signal interference between cells, LTE-A enhances channel quality of picocells by suspending the macrocell’s transmission in an almost blank subframe (ABS). Moreover, LTE-A adopts the discontinuous reception (DRX) method to let a user equipment (UE) adaptively disable its transceiver to save energy. However, existing studies view ABS and DRX as independent methods and address cell interference and energy saving separately. The paper points out that they have much in common, and develops a joint interference and power management (JIPM) mechanism to improve LTE-A performance by integrating ABS and DRX. Based on channel conditions and traffic demands of UEs, JIPM computes the amount of resources given to each UE and adjusts the parameters of ABS and DRX accordingly. Simulation results show that JIPM achieves higher energy efficiency and lower packet dropping of real-time flows, as compared with other schemes.

Index Terms—ABS, DRX, LTE-A.

1 INTRODUCTION

Thanks to the popularity of mobile phones and wireless networks, people desire to freely access the Internet today. Most operators thus provide long term evolution-advanced (LTE-A) services for high-speed communications. To guarantee seamless coverage while serving many user equipments (UEs) in hotspots, operators deploy picocell base stations (also called eNBs) within macrocells, and make them cooperate to form a heterogeneous network (HetNet). HetNet has many advantages such as flexible deployment [1] and load sharing [2]. It is also a trend to deploy numerous picocells to strengthen HetNets for the upcoming 5G epoch [3].

However, signal interference between a macrocell and its picocells is a big problem in HetNets, as the macrocell eNB emits much larger transmitted power than a picocell eNB. To conquer the problem, LTE-A uses the enhanced inter-cell interference coordination (eICIC) technique consisting of frequency-based and time-based methods. The frequency-based method asks macrocell and picocell eNBs to send data via different bands for interference avoidance. The time-based method aims to reduce the macrocell’s interference exerting on picocell UEs. Specifically, the macrocell eNB picks a number of subframes as almost blank subframes (ABSs) in a cycle, where it sends nothing but low-powered control signals. In this way, picocell UEs can receive data from their eNBs with very little interference from the macrocell eNB. Owing to its better utilization of the spectrum resource, the time-based method is more flexible and popular than the frequency-based one [4], [5], and it is also known as the ABS method.

How to reduce energy consumption of UEs is also critical. LTE-A deals with this issue by the discontinuous reception (DRX) method, which makes a UE alternate between sleeping and awake states. In the sleeping state, the UE turns off its transceiver to save energy (and it stops listening to the channel accordingly). In case that there are downlink data for the UE, the eNB keeps the data until the next time that the UE wakes up. Once the UE does not receive any data after passing a pre-defined number of such cycles, it can prolong the cycle’s length with a longer duration of the sleeping state to conserve more energy. When the UE gets data from the eNB (in the awake state), the cycle’s length is reset again. The DRX parameters such as cycle length and awake time are controlled by the eNB via sending management packets to the corresponding UEs. Thus, the eNB is aware of the current DRX state of each UE in its cell [6].

On the face of it, ABS and DRX seem to be two independent methods for interference and power management in LTE-A, respectively. In fact, they do share some similarities in terms of UEs’ behavior. Specifically, macrocell UEs cannot get data in an ABS, while a UE disables its transceiver in the sleeping state. However, existing studies aim to improve either the ABS or DRX methods individually. They may make macrocell UEs keep awake in ABSs, which wastes their energy. Even worse, picocell UEs may go to sleep when they have good channel quality (as the macrocell eNB stops data transmissions), which wastes the spectrum resource. Obviously, if the parameters of ABS and DRX can be co-adjusted properly such that UEs sleep and wake up at the right time (depending on their channel conditions), we can make a good balance between network throughput and energy consumption of UEs.

Therefore, this paper proposes a joint interference and power management (JIPM) mechanism to improve performance of LTE-A HetNets by integrating ABS and DRX through co-deciding their parameters. JIPM first synchronizes both ABS and DRX cycles. Based on the channel quality and traffic demand of each UE, it estimates the amount of resources required by the UE. Then, JIPM computes a suitable number of ABSs to balance macrocell and picocell loads, and decides the DRX
parameters accordingly. Through simulations, we show that the JIPM mechanism not only saves more energy of UEs but also increases network throughput, thereby greatly improving energy efficiency. Besides, JIPM can alleviate packet dropping of real-time flows as compared with other methods. Our contributions are to point out the benefits of integrating ABS and DRX and also develop an efficient mechanism to significantly improve performance of LTE-A HetNets.

This paper is outlined as follows: Section 2 briefly introduces LTE-A and Section 3 surveys related work. We propose the JIPM mechanism in Section 4. Then, Section 5 evaluates its performance. Finally, Section 6 concludes the paper and discusses future work.

2 LTE-A OVERVIEW

2.1 Resource Management

LTE-A divides the spectrum resource into a 2D array of physical resource blocks (PRBs) for management, where each PRB has a 0.5 ms duration and 180 kHz bandwidth. PRBs are exclusive when SISO or SU-MIMO techniques are used\(^2\), and no two UEs can share the same PRB. The eNB takes charge of allocating PRBs to UEs in each transmission time interval (TTI), whose length is 1 ms. If the channel has bandwidth of 1.4, 3, 5, 10, 15, or 20 MHz, the eNB offers 6, 15, 25, 50, 75, or 100 PRBs in a TTI, respectively. The amount of data sent by a PRB depends on its modulation and coding scheme (MCS). A more complex MCS lets the PRB send more data, but it needs less interference, and vice versa. Each UE reports the channel quality indicator (CQI) that reveals its channel condition to help the eNB choose MCS.

A UE may have multiple flows, each with its own QoS (quality of service) demand. LTE-A uses QoS class identifier (QCI) to describe two QoS parameters for a flow: delay budget and loss rate, which give the longest delay time that the flow can tolerate and the maximum allowable probability that its packets are dropped, respectively. LTE-A classifies flows into guaranteed-bit-rate (GBR) and non-GBR ones. GBR flows often support real-time services with strict delay demands (e.g., VoIP and video), while non-GBR flows are used for applications with loose deadlines (e.g., web browsing). Thus, GBR flows would have smaller QCIs and delay budgets than non-GBR ones.

2.2 Interference Management

A goal of HetNet is to share loads of macrocell eNBs by deploying picocells in their communication ranges. However, the transmitted power of a macrocell eNB is typically 46 dBm while that of a picocell eNB is at most 30 dBm [7]. Thus, UEs prefer connecting to macrocell eNBs due to their high reference signal received power (RSRP), which conflicts with the goal. To conquer this problem, LTE-A uses cell range expansion (CRE) by adding a small bias \(\psi\) to the RSRP values of picocells. Specifically, let \(E\) be the set of eNBs. A UE \(u_i\) selects an eNB \(e_j\) to connect based on

\[
e_j = \arg \max_{e_j \in E} (\xi_{i,j} + \psi)\text{ if } e_j \text{ is a picocell eNB},
\]

where \(\xi_{i,j}\) is \(u_i\)'s RSRP from \(e_j\). Fig. 1 gives an example, where picocell 2 virtually expands its range by CRE, so it can serve \(u_3\) and share the macrocell’s traffic load.

However, the macrocell eNB imposes large signal interference on picocell UEs, especially the UEs in the CRE area. Thus, LTE-A proposes the ABS method to reduce such interference. It divides the time axis into ABS cycles, during which some subframes are selected as ABSs. In an ABS, the macrocell eNB sends only low-powered control signals. Thus, picocell UEs incur little interference from it and can use complex MCSs to get data in a high rate. Fig. 1 shows the ABS method, where \(u_1\) and \(u_3\) improve their channel quality but \(u_2\) gets no data in ABSs. The ABS ratio (denoted by \(\delta\)) decides how many ABSs are selected in a cycle. As macrocell UEs have no throughput in ABSs, this ratio will affect network performance and fairness. LTE-A allows a macrocell eNB dynamically adjusting its \(\delta\) value to facilitate interference management.

2.3 Power Management

LTE-A uses DRX for power management, which cuts the time axis into DRX cycles, as shown in Fig. 2. In each cycle, a UE listens to the physical downlink control channel (PDCCH) for a while to check if there are data sent from the eNB. If not, the UE turns off its transceiver to save energy until the next cycle. In this case, the eNB will keep the UE’s data if necessary. There are two types of DRX cycles: short and long. If a UE does not receive any data after some short cycles, it changes to the long cycle to extend its sleeping time to conserve more energy.

In Fig. 2, the awake and sleeping time of a UE \(u_i\) is decided by six parameters: 1) \(DRX\) inactivity timer \((T^{IN}_i)\) indicates how many subframes that \(u_i\) should stay awake when it receives data from PDCCH, 2) short DRX cycle \((P^{SC}_i)\) is the length of a short cycle, 3) \(DRX\) short cycle timer \((T^{SC}_i)\) gives the deadline for \(u_i\) to change to a long cycle if it keeps receiving nothing from PDCCH, 4) long DRX cycle \((P^{LC}_i)\) is the length of a long cycle, and 5) on-duration timer \((T^{ON}_i)\) gives the number of subframes that \(u_i\) should stay awake in the beginning of a DRX cycle, and 6) DRX offset \((T^{OFF}_i)\) points out when a cycle begins. By sending radio resource configuration (RRC) messages to a UE which include the DRX parameters, the eNB can control the sleeping behavior and energy consumption of that UE.

---

2. SISO: single-input single-output, SU-MIMO: single-user multiple-input multiple-output
3 RELATED WORK

3.1 ABS Configuration

Several studies find the optimal ABS ratio $\delta$ and CRE bias $\psi$ with a full-buffer model, where each queue always has packets to be sent. The work [8] finds two sets of UEs to get data in ABSs and non-ABSs by dynamic programming, and derives $\delta$ to keep UEs’ fairness. Trabelsi et al. [9] search all possible ($\delta$, $\psi$) pairs, and pick the pair to maximize throughput. The study [10] assumes that UEs arrive to the network via a Poisson process and they have only FTP flows. It uses the response surface method, which is a stochastic optimization algorithm, to compute $\delta$ and $\psi$. However, not every flow follows the full-buffer model (e.g., VoIP and videos).

How to adjust $\delta$ to react to a dynamic network situation is also addressed. The work [11] raises $\delta$ with a probability when there are more UEs in picocells, but it does not consider traffic demands of UEs. In [12], both throughput and fairness maximization problems are modeled by sum-rate and product-traffic demands of UEs. In [12], both throughput and fairness maximization problems are modeled by sum-rate and product-traffic demands of UEs. In [14] to compute the ABS ratio, whose inputs contain UE number, macrocell throughput, and picocell channel quality. The work [15] formulates the problem of ABS configuration by a general-form consensus problem, and uses ADMM (alternating direction method of multiplier) to find a solution. However, the issue of packet latency is not discussed in these studies.

A few studies configure ABS with the goal of reducing GBR packet delay. The work [16] uses a genetic algorithm to allocate ABSs, whose fitness function takes cell throughput, noise interference, and video latency as inputs. Wang et al. [17] find a network-status indicator to check the suitability of using ABSs based on the capacity of each cell, and analyze queued data with different urgent levels to set the ABS ratio, so as to alleviate GBR packet dropping. However, [16] and [17] do not integrate ABS with DRX to balance between energy consumption and throughput of UEs.

3.2 DRX Configuration

Some studies set DRX parameters based on traffic loads or channel variations. The work [18] extends $T_{i}^{IN}$ to shorten sleeping time of UEs when their loads grow. Besides, if a UE’s channel condition keeps good for a while, it reduces $T_{i}^{IN}$ for energy saving. In [19], the UEs with larger CQIs are given with a smaller $T_{i}^{IN}$ value to conserve energy, as they can fast get data with more complex MCSs. However, both studies do not consider QoS demands of UEs.

How to configure DRX with delay concern is also discussed. The work [20] estimates the average delay by $(P_{i}^{LC} - T_{i}^{ON})/2$ and the power-saving ratio by $(P_{i}^{LC} - T_{i}^{ON})/(P_{i}^{LC} + T_{i}^{ON})$. It proposes two scenarios: 1) power-saving maximization with delay constraint and 2) delay minimization with power-saving constraint, to decide the values of $P_{i}^{LC}$, $T_{i}^{ON}$, and $T_{i}^{IN}$. In [21], a discrete-time Markov process is adopted to model the Internet traffic. By finding state transition probabilities from the process, it decides $P_{i}^{LC}$, $P_{i}^{SC}$, and $T_{i}^{SC}$ to reduce energy consumption of UEs and meet their delay demands. The study [22] adjusts $T_{i}^{ON}$, $T_{i}^{IN}$, and $T_{i}^{OFF}$ by referring to packet delay and CQI of each UE. If a UE has small packet delay and large CQI, it reduces both $T_{i}^{IN}$ and $T_{i}^{ON}$ while raising $T_{i}^{OFF}$ to save energy, as the UE can receive packets in a short time.

A number of studies combine DRX with resource scheduling. The work [23] sets DRX cycles (i.e., $P_{i}^{LC}$ and $P_{i}^{SC}$) of a UE as integer multiples of others to reduce the awake time of UEs due to resource competition. Then, the eNB allocates PRBs to the UEs whose packets will be dropped or timer $T_{i}^{IN}$ will expire in the next subframe, so as to catch packet deadlines. Tung et al. [24] set $P_{i}^{SC}$ as the minimum delay budget of a UE and $T_{i}^{ON}$ as the expected transmission time based on its CQI. Given the UE’s priority from a resource scheduling method, [24] adds a small value to the priority if the UE will go to sleep in the next subframe and the network load exceeds a threshold. The study [25] schedules multicast groups and configures DRX parameters such that the delay constraint of each multicast stream is not violated and total wake-up subframes of UEs can reduce. To do so, [25] decides the allocation order of multicast groups, including cycle lengths and offsets, to meet their delay demands. Then, it tunes both $T_{i}^{IN}$ and $T_{i}^{ON}$ of each UE in a group to reduce the wake-up time. The work [26] aims to support QoE (quality of experience) of GBR traffics and save energy of UEs. It proposes an opportunistic method to allocate resources to UEs, which considers multiple parameters including QoE requirement, channel condition, average throughput, buffer length, DRX status, and GBR/non-GBR flows.

However, the above studies neither consider the HetNet scenario nor address signal interference when UEs are awake. It motivates us to develop the JIPM mechanism by integrating both ABS and DRX for joint management of interference and power in LTE-A HetNets.

4 THE PROPOSED JIPM MECHANISM

4.1 Network Model

In the ABS method, each macrocell decides its ratio $\delta$. For ease of explanation, our discussion aims at a basic HetNet with some picocells enclosed by one macrocell, and the result can be easily extended to a general HetNet with multiple macrocells. Let us denote by $E$ and $E_i$ the sets of all and picocell eNBs in the basic HetNet, respectively. All eNBs operate on the same frequency band and allot non-sharable PRBs to UEs by using SU-MIMO. Besides, each UE associates with only one eNB, and we denote by $U_j$ the set of UEs linking to eNB $e_j \in E$. A UE may generate GBR or non-GBR flows, where GBR flows have stringent delay requirements. To save their energy, we consider using long DRX cycles for UEs.

Given traffic demands of UEs, our problem asks how to decide the ABS ratio $\delta$ for the HetNet and also DRX cycle length $P_{i}^{LC}$, on-duration timer $T_{i}^{ON}$, and DRX offset $T_{i}^{OFF}$ of each UE such that energy efficiency is maximized while GBR packet dropping is minimized. Specifically, energy efficiency is defined by the ratio of HetNet throughput to energy consumption of UEs. Besides, a GBR packet will be dropped if it is overdue. Table 1 summarizes our notations.
TABLE 1: Summary of notations.

<table>
<thead>
<tr>
<th>notation</th>
<th>definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>E, E_p</td>
<td>the set of all/picocell eNBs in a basic HetNet</td>
</tr>
<tr>
<td>U_t</td>
<td>the set of UEs associating with eNB e_t</td>
</tr>
<tr>
<td>P_{ABS}, P_{LC}</td>
<td>ABS cycle length and DRX long cycle length</td>
</tr>
<tr>
<td>T_{ON}, T_{OFF}</td>
<td>on-duration timer and DRX offset of UE u_i</td>
</tr>
<tr>
<td>Γ_{i,j}^A, Γ_{i,j}^O</td>
<td>SINR of UE u_i from eNB e_j in a non-ABS/ABS</td>
</tr>
<tr>
<td>Q_i, Q_A</td>
<td>CQI of UE u_i in a non-ABS/ABS</td>
</tr>
<tr>
<td>F(b_i, Q_i)</td>
<td>a function to find the number of PRBs to send b_i data bits with CQI Q_i</td>
</tr>
<tr>
<td>u_i^{G.O}, u_i^{G.A}</td>
<td>the number of PRBs to send UE u_i’s GBR data in a cycle with non-ABSs/ABSs</td>
</tr>
<tr>
<td>u_i^{NG.O}, u_i^{NG.A}</td>
<td>the number of PRBs to send UE u_i’s non-GBR data in a cycle with non-ABSs/ABSs</td>
</tr>
<tr>
<td>N_j</td>
<td>available PRBs provided by eNB e_j in a cycle</td>
</tr>
<tr>
<td>L_j</td>
<td>traffic load of eNB e_j for GBR or non-GBR flows</td>
</tr>
<tr>
<td>γ</td>
<td>the number of ABSs in a cycle (ABS ratio: δ = γ/P_{ABS})</td>
</tr>
</tbody>
</table>

2) Resource estimation: With CQI and cycle length, we estimate how many PRBs are required to satisfy the traffic demand of each UE.

3) ABS setting: Based on the amount of traffics in cells, we find the ABS ratio δ to raise throughput and balance loads, and also allocate PRBs to UEs.

4) DRX setting: Both T_{ON} and T_{OFF} of UEs are then decided by their allocated resources.

4.2.1 CQI Evaluation

Let D_{min} be the minimum delay budget of all flows owned by a UE u_i. Then, both ABS cycle length P_{ABS} and DRX cycle length P_{LC} are set by

\[ P_{ABS} = P_{LC} = \min_{u_i \in D} \{D_{min}\}, \text{ where } D = \bigcup_{j \in E} U_j. \]  

(2)

In other words, the lengths of these cycles are decided by the most stringent delay demand of flows in the network, so as to catch their packet deadlines.

We then evaluate the signal-to-interference-plus-noise ratio (SINR) of each UE. Suppose that eNB e_j sends data to UE u_i over subchannel c_k. Thus, u_i’s SINR in a non-ABS is

\[ \Gamma_{i,j,k}^O = \frac{S(u_i, e_j, c_k)}{\varepsilon N_0 B_k + \sum_{e_h \in E, e_h \neq e_j} S(u_i, e_h, c_k)}, \]  

(3)

where \( S(u_i, e_j, c_k) \) is the strength of power received by u_i from e_j over c_k, which depends on e_j’s transmitted power and the distance between u_i and e_j. In Eq. (3), the calculation of SINR \( \Gamma_{i,j,k}^O \) considers not only the interference \( S(u_i, e_h, c_k) \) from other eNBs but also the effect of environmental noise \( \varepsilon N_0 B_k \), where \( \varepsilon \) is a constant noise figure, \( N_0 \) is the power spectral density of noise (usually set to -174 dBm/Hz), and \( B_k \) is the bandwidth of subchannel c_k (i.e., 15 kHz). On the other hand, there are two cases to compute u_i’s SINR in an ABS:

- u_i is a macrocell UE: Since the macrocell eNB does not send user data, its transmitted power can be ignored. By Eq. (3), the SINR \( \Gamma_{i,j,k}^A \) will be very small.
- u_i is a picocell UE: Only the interference from other picocells should be considered. Thus, its SINR is computed by

\[ \Gamma_{i,j,k}^A = \frac{S(u_i, e_j, c_k)}{\varepsilon N_0 B_k + \sum_{e_h \in E, e_h \neq e_j} S(u_i, e_h, c_k)}, \]  

(4)

Given \( u_i \)'s SINR over each subchannel c_k, its overall SINR in a non-ABS can be estimated by the exponential effective SINR mapping (EESM) approach [27]:

\[ \Gamma_{i,j}^O = -\beta \ln \frac{1}{m_C} \sum_{k=1}^{m_C} \exp(-\gamma \Gamma_{i,j,k}^O / \beta), \]  

(5)

where \( \beta \) is a coefficient (usually set to 1), \( m_C \) is the number of subchannels, and \( \exp(x) \) is the exponential function. Similarly, the overall SINR \( \Gamma_{i,j}^A \) of u_i in an ABS is computed by Eq. (5) through replacing \( \Gamma_{i,j,k}^O \) with \( \Gamma_{i,j,k}^A \).

Table 2 lists the MCS supported by each CQI and its minimum required SINR [28], where QPSK and QAM are the acronyms of “quadrature phase-shift keying” and “quadrature amplitude modulation”, respectively, and the numbers in brackets indicate code rates. Given Table 2, we can compute the CQI for u_i in a non-ABS (denoted by \( Q^O \)) by finding the maximum CQI whose SINR is no larger than \( \Gamma_{i,j}^O \) from Table 2. The CQI \( Q^A \) of u_i in an ABS can be found by \( \Gamma_{i,j}^A \) in the same way. Lemma 1 analyzes the time complexity of our CQI evaluation scheme.

**Lemma 1.** Given \( \tilde{m}_U \) UEs, \( \tilde{m}_E \) eNBs, and \( \tilde{m}_C \) subchannels, the CQI evaluation scheme has time complexity of \( O(\tilde{m}_U \tilde{m}_E \tilde{m}_C) \) in the worst case.

**Proof:** We first compute \( P_{ABS} \) and \( P_{LC} \) by Eq. (2), which finds the minimum delay budget of all flows. Obviously, it takes \( O(\tilde{m}_U) \) time, where \( \tilde{m}_U \) is the number of flows. Then, Eq. (3) finds the SINR of a UE over a subchannel, which spends \( O(\tilde{m}_E) \) time as we should consider the power \( S(u_i, e_h, c_k) \) from every eNB e_h. Since there are \( \tilde{m}_U \) UEs and \( \tilde{m}_C \) subchannels, it takes time of \( \tilde{m}_U \tilde{m}_C \cdot O(\tilde{m}_E) \) to find \( \Gamma_{i,j,k}^O \) for all UEs in a non-ABS. On the other hand, we need not find SINRs of macrocell UEs in an ABS. Besides, it spends time of \( \tilde{m}_U \cdot \tilde{m}_C \cdot O(\tilde{m}_E) \) to find \( \Gamma_{i,j,k}^A \) of all picocell UEs in an ABS (since Eq. (4) considers the power only from picocell eNBs), where \( \tilde{m}_U \) and \( \tilde{m}_C \) denote the numbers of picocell UEs and eNBs, respectively.

After getting all \( \Gamma_{i,j,k}^O \) (respectively, \( \Gamma_{i,j,k}^A \)) values, we use EESM in Eq. (5) to find \( \Gamma_{i,j}^O \) (respectively, \( \Gamma_{i,j}^A \)) for each UE, which takes \( O(\tilde{m}_C) \) time to do the calculation. As there are \( \tilde{m}_U \) UEs, it spends time of \( 2\tilde{m}_U \cdot O(\tilde{m}_C) \) to estimate the overall SINR of each UE. Finally, we pick a CQI for each UE in non-ABSs and ABSs by consulting Table 2. Since Table 2 contains only 15 CQIs, it takes constant time to do the consultation. Thus, finding CQIs for all UEs takes \( O(20\tilde{m}_U) \) time.

To sum up, the CQI evaluation scheme has time complexity of \( O(\tilde{m}_U \tilde{m}_E \tilde{m}_C) \) + \( O(2\tilde{m}_U \tilde{m}_C) \cdot O(\tilde{m}_E) \).
2\(\tilde{m}_U \cdot O(\tilde{m}_C) + 2O(\tilde{m}_U)\). Since \(\tilde{m}_{U,P} \leq \tilde{m}_U\), \(\tilde{m}_{E,P} \leq \tilde{m}_E\), and \(\tilde{m}_F \leq \tilde{m}_U \tilde{m}_C\tilde{m}_E\) (unless each UE has more than \(\tilde{m}_C\tilde{m}_E\) flows, but it is infeasible), we can simplify the above equation to \(O(\tilde{m}_U\tilde{m}_E\tilde{m}_C)\) and verify this lemma.

### 4.2.2 Resource Estimation

This stage estimates the amount of resources (in PRBs) required by each UE. LTE-A offers three tables [29] to find the number of data bits that a UE can get based on its CQI and PRBs:

- **CQI-MCS mapping table**: This table provides the translation of a CQI \(Q_i\) into an MCS index \(I_{MCS}\). We use a function \(F_1(Q_i) = I_{MCS}\) to represent the translation.

- **MCS-TBS mapping table**: It translates \(I_{MCS}\) into a TBS (transport block size) index \(I_{TBS}\), where TBS gives the number of bits that a PRB can carry. This translation is described by a function \(F_2(I_{MCS}) = I_{TBS}\).

- **TBS-bit mapping table**: Given \(I_{TBS}\) and \(n_i\) PRBs, the table returns how many bits (i.e., \(b_i\)) carried by these PRBs. A function \(F_3(I_{TBS}, n_i) = b_i\) is used to indicate the translation.

Through these tables, we can compute the number of PRBs used to send data with \(b_i\) bits:

\[
F(b_i, Q_i) = \arg \min_{n_i} \{ F_3(F_2(F_1(Q_i)), n_i) \geq b_i \}. \tag{6}
\]

Eq. (6) means to use the minimum PRBs that carry at least \(b_i\) bits based on CQI \(Q_i\).

Suppose that each flow \(f_{i,j}\) of UE \(u_i\) has data rate of \(r_{i,j}\) (in bps). We compute the number of PRBs to satisfy \(u_i\)'s GBR demands in a cycle with non-ABSs:

\[
n^{G,O}_i = \sum_{f_{i,j}} \{ F(r_{i,j} \cdot P_i^{LC}/1000, Q_i^O) \mid f_{i,j} \text{ is GBR} \}, \tag{7}
\]

the number of PRBs to satisfy \(u_i\)'s GBR demands in a cycle with ABSs:

\[
n^{G,A}_i = \sum_{f_{i,j}} \{ F(r_{i,j} \cdot P_i^{LC}/1000, Q_i^A) \mid f_{i,j} \text{ is GBR} \}, \tag{8}
\]

the number of PRBs to meet \(u_i\)'s non-GBR demands in a cycle with non-ABSs:

\[
n^{NG,O}_i = \sum_{f_{i,j}} \{ F(r_{i,j} \cdot P_i^{LC}/1000, Q_i^O) \mid f_{i,j} \text{ is non-GBR} \}, \tag{9}
\]

and the number of PRBs to meet \(u_i\)'s non-GBR demands in a cycle with ABSs:

\[
n^{NG,A}_i = \sum_{f_{i,j}} \{ F(r_{i,j} \cdot P_i^{LC}/1000, Q_i^A) \mid f_{i,j} \text{ is non-GBR} \}. \tag{10}
\]

In Eqs. (7)–(10), the unit of \(P_i^{LC}\) is ms, so \((r_{i,j} \cdot P_i^{LC}/1000)\) gives the number of data bits produced by \(f_{i,j}\) in a cycle. Lemma 2 analyzes the time complexity of the resource estimation scheme.

### Lemma 2

Given \(\tilde{m}_F\) flows, the resource estimation scheme spends at most \(20(\tilde{m}_F)\) time.

**Proof**: Each mapping table can be implemented by an array. In this way, we can easily retrieve the value of any desired entry from a table by giving its index(es) (i.e., input(s)). Thus, the calculation of \(F_3(F_2(F_1(Q_i)), n_i)\) takes only constant time. Besides, it is trivial to do the comparison in Eq. (6). So, the calculation of function \(F(b_i, Q_i)\) is \(O(1)\). Observing from Eqs. (7)–(10), we execute function \(F(b_i, Q_i)\) twice for each flow (one for the non-ABS case and the other for the ABS case). Since there are \(\tilde{m}_F\) flows, the resource estimation scheme spends time of \(20(\tilde{m}_F)\) in the worst case.

### 4.2.3 ABS Setting

Let \(\gamma\) be the number of ABSs in a cycle. We set \(\gamma = 0\) and estimate the load of each eNB in terms of GBR flows. Then, \(\gamma\) is gradually increased until the load meets some conditions, and we decide the amount of resources given to each GBR flow. If some PRBs remain, they are allocated to non-GBR flows. Specifically, the ABS setting scheme contains five steps below.

**Step 1**: We compute the number of available PRBs given by each eNB \(e_j\) in a cycle:

\[
N_j = \begin{cases} 
\tilde{N}(B_j) \cdot P^{ABS} & \text{if } e_j \in \mathcal{E}_p \\
\tilde{N}(B_j) \cdot \max\{P^{ABS} - \gamma, 0\} & \text{otherwise}, 
\end{cases} \tag{11}
\]

where \(\tilde{N}(B_j)\) is the number of PRBs that \(e_j\) offers in a subframe, which depends on its channel bandwidth \(B_j\) (referring to Section 2.1). If \(e_j\) is the macrocell eNB, it can allocate PRBs only in \(P^{ABS} - \gamma\) non-ABSs. Since it is impossible to use more than \(P^{ABS}\) ABSs in a cycle (i.e., \(\gamma > P^{ABS}\)), we take the term \(\max\{P^{ABS} - \gamma, 0\}\) in Eq. (11). Then, the load of a macrocell eNB \(e_m\) with respect to GBR flows is calculated by

\[
L_m = \begin{cases} 
0 & \text{if } U_m = \emptyset \\
\sum_{u_i \in U_m} n^{G,O}_i / N_m & \text{if } N_m > 0 \\
L_{\text{max}} & \text{otherwise.}
\end{cases} \tag{12}
\]

In Eq. (12), if \(e_m\) serves no UE (i.e., \(U_m = \emptyset\)), its load is obviously zero. On the other hand, if \(N_m = 0\) (this case occurs when \(\gamma \leq P^{ABS}\) in Eq. (11)), \(L_m\) is set to the maximum limit \(L_{\text{max}}\) of load, because \(e_m\) cannot send user data in the current cycle.

For a picocell eNB \(e_j\), its load with respect to GBR flows is computed by

\[
L_j = \left( \sum_{u_i \in U_j} n^{G,O}_i - N_j^{ABS} \right) / N_j. \tag{13}
\]

Since \(e_j\) can benefit from ABSs to speed up data transmissions (as channel quality is improved), we deduct \(N_j^{ABS}\) from its required number of PRBs in Eq. (13), where

\[
N_j^{ABS} = \sum_{u_i \in U_j} \left( \left( n^{G,O}_i / n^{G,A}_i \right) \times \alpha_i^A \right) - \alpha_i^A. \tag{14}
\]

Here, \(u_i\) is given with \(\alpha_i^A\) ABS-based PRBs. For example, suppose that \(u_i\) spends 120 and 20 PRBs to complete sending its GBR data in non-ABSs and ABSs, respectively (i.e., \(n^{G,O}_i = 120\) and \(n^{G,A}_i = 20\)). If \(u_i\) is given with 10 ABS-based PRBs, it requires extra 60 non-ABS-based PRBs to finish sending its GBR data (i.e., \(u_i\) totally spends \(10 + 60 = 70\) PRBs). Thus, comparing with the case where \(u_i\) uses only non-ABSs, it can save \((120 - 70) = 50\) PRBs. From Eq. (14), we have \(120 / 10 = 120 / 20 = 10\), which derives the same result.

Moreover, \(\alpha_i^A\) is estimated by

\[
\alpha_i^A = \left[ \gamma \tilde{N}(B_j) / |U_j^G| \right], \tag{15}
\]

which means that we give the average number of ABS-based PRBs to each UE with GBR flows (denoted by \(U_j^G\)). In case that \(\gamma \tilde{N}(B_j)\) is not divisible by \(|U_j^G|\) in Eq. (15), we give one ABS-based PRB to each UE in a round-robin manner.

3. An ABS-based PRB is a PRB allocated by one picocell eNB in an ABS.
Step 2: We then determine whether to add ABSs by
\[ \sum_{e_j \in E_p} L_j / |E_p| > 1. \]  
Eq. (16) implies that most picocell eNBs have not enough resources to meet their GBR demands, so it is better to use more ABSs to improve picocell throughput. Therefore, we iteratively increase $\gamma$ by one and recompose each eNB's load by Eqs. (12) and (13), until 1) Eq. (16) is violated, 2) $\gamma$ reaches $P^{ABS}$ (i.e., no more ABSs can be added), or 3) the condition holds:
\[ L_m > 0 \text{ and } \sum_{e_j \in E_p} L_j / |E_p| < L_m. \]  
Eq. (17) means that the macrocell has a heavier load (i.e., $L_m$) than picocells due to the decrease of $N_m$ in Eq. (12). Thus, we stop adding ABSs to balance loads of the macrocell and its picocells.

Step 3: When $L_m \leq 1$, the macrocell eNB $e_m$ has enough resources to satisfy all GBR demands. Thus, $e_m$ gives each UE its desired number $n_i^{G,O}$ of PRBs and updates $N_m$ by $N_m - \sum_{u_i \in U_m} n_i^{G,O}$ (i.e., $N_m$ is the number of residual PRBs).

When $L_m > 1$, we use a top-50%-first rule to assign PRBs to each UE as follows: Let $Q^{O}_{\text{avg}}$ be the average CQI. UEs are divided into good-channel and bad-channel groups (denoted by $G_g$ and $G_b$, respectively). If a UE $u_i$ satisfies the condition $Q_{\text{avg}}(u_i) \times L_j \leq 1$, where $Q_{\text{avg}}$ is a lower-bound threshold for CQI, it is added to $G_g$. Otherwise, $u_i$ is added to $G_b$.

Then, we first allocate PRBs to the UEs in $G_g$. Specifically, we repeat the following operations: 1) pick a UE $u_i$ with the largest CQI from $G_g$, 2) allocate $n_i^{G,O}$ PRBs to $u_i$, and 3) set $N_m = N_m - n_i^{G,O}$, until $N_m = 0$ or $G_g = \emptyset$. If $N_m > 0$ (i.e., some PRBs remain), we proportionally distribute them among the UEs in $G_b$. Specifically, for each $u_i \in G_b$, we give it $[(n_i^{G,O} / \sum_{u_j \in G_b} n_j^{G,O}) \times N_m]$ PRBs. Remark 2 discusses the idea behind the top-50%-first rule.

Step 4: A picocell eNB $e_j$ has sufficient resources to meet all GBR demands if $L_j \leq 1$. In this case, $e_j$ gives the required number of PRBs to each UE as follows:

- Sort all UEs by their $Q^{A}_i$ values in a decreasing order.
- Based on the order, we iteratively allocate a number $n_i^{G,A}$ of ABS-based PRBs to a UE $u_i$ to meet its GBR demand, until we use up ABS-based PRBs. However, if there are not enough ABS-based PRBs to satisfy $u_i$'s demand, we adopt Eq. (14) to compute the number of extra non-ABS-based PRBs allocated to $u_i$.
- Then, we give a number $n_j^{G,O}$ of non-ABS-based PRBs to each UE in $U_j$, whose GBR demand has not been satisfied yet. Also, $N_j$ is updated by the number of residual PRBs owned by $e_j$.

Otherwise, $e_j$ cannot support GBR demands of all UEs. Thus, $e_j$ adopts the top-50%-first rule to assign PRBs to its UEs, and then sets $N_j = 0$.

Step 5: If $N_j \neq 0$ for some eNBs (i.e., they still have available PRBs), we check whether $N_j$ is large enough to meet $n_i^{G,O}$ or $n_i^{G,A}$ demands of non-GBR flows in their cells. If so, these UEs are given with PRBs as they require (i.e., similar to the above method to deal with GBR flows). Otherwise, we use the top-50%-first rule to assign PRBs to UEs for their non-GBR flows.

By the above steps, the ABS ratio will be $\delta = \gamma / P^{ABS}$. Remark 1 discusses why the ABS setting scheme can find a suitable $\gamma$ value to balance loads of a macrocell and its picocells with respect to GBR flows. Then, Lemma 3 analyzes the time complexity of this scheme.

**Remark 1** (Load-balancing property). Our ABS setting scheme can balance GBR loads of a macrocell and its picocells (i.e., minimizing the difference between their GBR loads). Specifically, there are three possible cases for a macrocell eNB $e_m$ and its picocell eNBs $e_j \in E_p$ in the beginning: 1) $L_m = 0$, 2) $\sum_{e_j \in E_p} L_j / |E_p| < L_m$, and 3) $\sum_{e_j \in E_p} L_j / |E_p| \geq L_m$.

For case 1, since $e_m$ has no GBR load, it does not matter to balance loads of $e_m$ and picocell eNBs. However, the ABS setting scheme still increases $\gamma$ until either $\sum_{e_j \in E_p} L_j / |E_p| \leq 1$ (i.e., each picocell eNB can satisfy all GBR demands) or $\gamma = P^{ABS}$ (i.e., no ABSs can be further added). In this way, we guarantee that the overall GBR throughput is maximized.

Case 2 implies that $e_m$ has a heavier GBR load than its picocell eNBs, even if there is no ABS in the current cycle. In this case, increasing $\gamma$ will raise the gap between the load of $e_m$ and those of its picocell eNBs, because $L_m$ increases (referring to Eqs. (11) and (12)) while $\sum_{e_j \in E_p} L_j / |E_p|$ decreases (referring to Eq. (13)) when $\gamma$ grows. In fact, the condition of Eq. (17) holds in the beginning, which makes the ABS setting scheme keep $\gamma = 0$ and minimize the gap accordingly.

For case 3, the ABS setting scheme starts from $\gamma = 0$ and increases $\gamma$ by one in each iteration of step 2. When the GBR loads of $e_m$ and its picocells become balanced (i.e., the gap is minimized), the condition of Eq. (17) will also become true. In this case, the ABS setting scheme stops increasing $\gamma$, which gets a suitable $\gamma$ value to balance cell loads. □

**Lemma 3.** Let $\tilde{m}_u$, $\tilde{m}_i$, and $\tilde{m}_F$ be the numbers of UEs, eNBs, and flows, respectively. Then, the worst-case time complexity of the ABS setting scheme is $O(P^{ABS} \cdot \tilde{m}_F \lg \tilde{m}_F) + O(\tilde{m}_F \lg \tilde{m}_F).

Proof: The ABS setting scheme can be divided into two parts. Part 1 repeats both steps 1 and 2 to calculate $\gamma$. Part 2 contains steps 3, 4, and 5 for each eNB to allocate PRBs to its UEs. Below, we analyze the time complexity of each part.

In an iteration of part 1, we use Eq. (12) to find the GBR load of each macrocell, which spends time of $\tilde{m}_E \cdot \tilde{m}_U$ for each UE. We denote the numbers of macrocell eNBs and UEs, respectively. Similarly, we use Eq. (13) to estimate the GBR load of each picocell, which takes time of $\tilde{m}_{E,P} \cdot \tilde{m}_{U,P}$, where $\tilde{m}_{E,P}$ and $\tilde{m}_{U,P}$ are the numbers of picocell eNBs and UEs, respectively. Then, we check whether to add ABSs by Eq. (16), which requires $O(\tilde{m}_{E,P})$ time. The three conditions in step 2 actually take constant time (including the calculation of Eq. (17), as we already know the value of $\sum_{e_j \in E_p} L_j / |E_p|$ from Eq. (16)). Thus, one iteration takes time of $\tilde{m}_E \cdot \tilde{m}_U + \tilde{m}_{E,P} \cdot \tilde{m}_{U,P} = O(\tilde{m}_E \cdot \tilde{m}_U + \tilde{m}_{E,P} \cdot \tilde{m}_{U,P})$. The worst case occurs when we increase $\gamma$ from zero to $P^{ABS}$. In other words, there are at most $P^{ABS}$ iterations. So, the time complexity of part 1 is $O(P^{ABS} \cdot \tilde{m}_F \lg \tilde{m}_F)$.

For part 2, we observe that the top-50%-first rule dominates the computational time of steps 3, 4, and 5. Thus, the worst case is that all flows belong to GBR and an eNB has to use this rule to allocate PRBs to every flow. In the top-50%-first rule, we sort flows by their CQIs, which spends $O(\tilde{m}_F \lg \tilde{m}_F)$ time. Then, the eNB iteratively picks one flow and gives PRBs to it.
This operation takes \(O(\log n_F)\) time. Thus, the time complexity of part 2 is \(O(\log n_F \log n_U) + O(\log n_F) = O(\log n_F \log n_U)\).

To sum up, the time complexity of the ABS setting scheme is \(O(P_{ABS} \log n_U) + O(\log n_F \log n_U)\), which verifies this lemma.

**Remark 2** (The idea behind the top-50%-first rule). Both max-CQI and proportional fair (PF) are two classic scheduling policies in LTE-A [30]. Max-CQI always gives each PRB to the UE with the best channel quality (i.e., the largest CQI). Due to its greedy nature, max-CQI can achieve the highest throughput in theory. However, other UEs with smaller CQIs may not get any resource (i.e., starvation). On the contrary, PF seeks to allocate PRBs in a fair manner, so UEs may get PRBs proportionally to their demands. However, network throughput will significantly degrade if some UEs have pretty low CQIs. Based on these observations, our top-50%-first rule combines the advantages of both max-CQI and PF policies. In particular, it divides UEs into good-channel \(G_g\) and bad-channel \(G_b\) groups. Group \(G_g\) contains the first 50% of UEs in terms of channel quality. Thus, our rule adopts the max-CQI policy to allocate PRBs to these UEs, which can greatly improve throughput by using fewer PRBs. On the other hand, if there are residual PRBs, they are given to other UEs (i.e., \(G_b\)) following the PF policy, so as to avoid starving some UEs. Note that when most UEs have bad channel quality, it is not suitable to adopt the max-CQI policy [31]. The reason is that each UE spends more PRBs to carry its data. Since the total number of PRBs is fixed, max-CQI starves more UEs. In this case, PF is a better choice. Therefore, we apply the lower-bound threshold \(Q_{th}\) to the top-50%-first rule. When \(Q_U \geq Q_{th}\) but \(Q_U < Q_{th}\), UE \(u_i\) is added to the bad-channel group \(G_b\), which is allocated with PRBs by the PF policy. The work [32] points out that a PRB carries relatively more data bits by using QAM than using QPSK. Thus, we suggest setting \(Q_{th}\) to the lowest CQI capable of using QAM (i.e., \(Q_{th} = 7\) from Table 2). Based on the above design, the top-50%-first rule can keep high throughput even if there are few PRBs available (thanks to the max-CQI property). Besides, when there are more PRBs offered by an eNB, this rule can improve UE fairness by taking the PF policy.

4.2.4 DRX Setting

After determining the number of PRBs given to each UE \(u_i\), we can find its PRB deployment and also \(T^{ON}\) and \(T^{OFF}\) values. To better utilize the spectrum resource and let UEs save more energy, we should compact PRBs allocated to them. There are two cases to be discussed. For the macrocell case, we start deploying PRBs from the beginning of an ABS cycle, and place a number \((n_{G,O}^i + n_{NG,O}^i)\) of PRBs for each UE in sequence. Fig. 3(a) gives an example, where \(P_{ABS} = 15\) ms and \(N(B_j) = 10\). Suppose that UEs \(u_{12}, u_{13}, u_{14}, u_5\) are given with 12, 14, 23, 36, and 8 PRBs by the ABS setting scheme, respectively. From the deployment of PRBs in Fig. 3(a), we can easily decide both \(T^{ON}\) and \(T^{OFF}\) for each UE.

For the picocell case, we first deploy PRBs for the UEs which use only non-ABS-based PRBs (i.e., \(n_{G,A}^i + n_{NG,A}^i = 0\) but \(n_{G,O}^i + n_{NG,O}^i > 0\)) by the method in the macrocell case. Then, we deploy PRBs for the UEs which use both non-ABS-based and ABS-based PRBs (i.e., \(n_{G,A}^i + n_{NG,A}^i > 0\) and \(n_{G,O}^i + n_{NG,O}^i > 0\)). Finally, we deploy PRBs for the UEs which use merely ABS-based PRBs (i.e., \(n_{G,A}^i + n_{NG,A}^i > 0\) but \(n_{G,O}^i + n_{NG,O}^i = 0\)). In this way, we can guarantee that all UEs are synchronized by the same ABS cycle.

However, the above deployment of PRBs is not optimal, since some UEs have to keep awake for a longer \(T^{ON}\) time, thereby spending more energy. Fig. 3 gives two examples. When we deploy PRBs based on the sequence of UEs, as shown in Fig. 3(a), all UEs wake up for 14 subframes. In fact, one can use the deployment of PRBs in Fig. 3(b) to minimize the awake time to 12 subframes. We formulate this problem as a PRB deployment with the minimum awake subframes (PDMAS) problem in Definition 1, and Theorem 1 shows that it is NP-hard.

**Definition 1.** Given the number of PRBs \(N(B_j)\) provided by eNB \(e_j\) in a subframe and the number of PRBs \(n_i\) assigned to each UE \(u_i\), the PDMAS problem asks how to arrange the PRBs allocated to UEs in a cycle, such that the sum of \(T^{ON}\) values of all UEs in \(U_j\) is minimized.

**Theorem 1.** The PDMAS problem is NP-hard.

**Proof:** Observing from Fig. 3, when \(n_i\) is divisible by \(N(B_j)\) for a UE \(u_i\), the best way to minimize its \(T^{ON}\) value is to give it the whole PRBs in \(n_i/N(B_j)\) subframes. Thus, \(u_i\) will not share the PRBs in a subframe with others. In general, when a subset \(U_j \subset U\) of UEs meet the condition of \(\sum_{u_i \in U_j} n_i \mod N(B_j) = 0\), we can allocate the whole PRBs in \(\sum_{u_i \in U_j} n_i/N(B_j)\) subframes to minimize their \(T^{ON}\) values. Fig. 3(b) gives an example, where we put the PRBs of \(u_1\) and \(u_5\) in subframes 0 and 1, and also the PRBs of \(u_2\) and \(u_4\) in subframes 2–6. Thus, if we find all of such subsets \(U_j\), the sum of \(T^{ON}\) values of all UEs must be the minimum.

Since any two UEs in different subsets will not share PRBs in the same subframe, the PDMAS problem will be equivalent to the problem of finding the largest subset \(U_j\) of UEs such that \(\sum_{u_i \in U_j} n_i \mod N(B_j) = 0\). Thus, we formulate its decision problem as follows: Given \(n_i\) of each UE in \(U_j\), there is any subset of UEs such that \(\sum_{u_i \in U_j} n_i = kN(B_j)\), where \(k \in N\) is a constant. In fact, there exists a similar NP-complete problem, called the subset-sum problem: Given a set \(X\) of integers and also a target integer \(t\), can we find a non-empty subset from \(X\) whose sum is equal to \(t\)? For example, given \(X = \{3, 1, 7, 8, 10\}\) and \(t = 16\), the solution is \(\{1, 7, 8\}\). Thus,
we construct a PDMAS problem instance to do the reduction from the subset-sum problem. Suppose that \( n_i < N(B_i) \) for any UE in \( \mathcal{U}_i = \{u_1, u_2, \ldots, u_m\} \). Then, we construct a set \( X = \{x_1, x_2, \ldots, x_m\} \) such that \( x_i = n_i \) for \( i = 1..m \). In this way, if we find a solution to the subset-sum problem, we can derive the solution subset \( \mathcal{U}_i \) to the PDMAS problem, and vice versa. The above reduction takes polynomial time, so the PDMAS problem is NP-hard.

To solve the subset-sum problem, we adopt the linear approximation algorithm in [33], whose idea is to split the set \( X \) into two subsets that consist of large and small integers. It selects large integer(s) and then adds small integers based on a greedy approach, which checks small integers in any order and inserts each new one if the sum does not exceed \( t \). Given the set \( X \) and the target integer \( t \), this algorithm contains four steps to find the solution list \( \mathcal{L} \):

1. Let \( s_1 \) be the sum of the three minimum integers greater than \( \lceil t/4 \rceil \). If \( s_1 \leq t \), we set \( s = s_1 \), move the three integers from \( X \) to \( \mathcal{L} \), and go to step 4.
2. Let \( s_2 \) be the sum of the minimum integer greater than \( \lceil t/4 \rceil \) and the minimum integer greater than \( \lceil t/2 \rceil \). If \( s_2 \leq t \), we set \( s = s_2 \), move the two integers to \( \mathcal{L} \), and go to step 4.
3. Let \( s_3 \) be the sum of the two largest integers which do not exceed \( \lceil t/2 \rceil \) and also \( s_3 \) be the largest integer. We set \( s = \max\{s_3, s_4\} \), and move the corresponding integer(s) to \( \mathcal{L} \) based on the selection of max operation.
4. Iteratively pick integers from \( X \) by the greedy approach and move them to \( \mathcal{L} \), until their sum reaches \( (t - s) \).

Let us use the term \( \mathcal{L} = \text{LA}(X, t) \) to denote the above algorithm. Then, we propose a method to solve the PDMAS problem as follows:

1. For each UE \( u_i \in \mathcal{U}_j \), we check if \( n_i \mod N(B_j) = 0 \). If so, we give \( n_i \) PRBs to \( u_i \) and remove it from \( \mathcal{U}_j \).
2. Build a set \( \mathcal{X} \) of integers from \( \mathcal{U}_j \), where each integer \( x_i \in \mathcal{X} \) is computed by \( (n_i \mod N(B_j)) \) corresponding to each UE \( u_i \in \mathcal{U}_j \). Then, we run LA(\( \mathcal{X}, N(B_j) \)) and get the list \( \mathcal{L} \). If \( \sum_{x_i \in \mathcal{X}} x_i = \tilde{N}(B_j) \), we deploy PRBs for the corresponding UEs by referring to \( \mathcal{L} \) and remove these UEs from \( \mathcal{U}_j \). Otherwise, we go to step 4.
3. Repeat step 2, until \( |\mathcal{U}_j| < 3 \) (since it is a trivial case when there exist only two UEs).
4. Deploy PRBs for residual UEs in \( \mathcal{U}_j \) in sequence.

Fig. 3 gives an example, where \( \tilde{N}(B_j) = 10 \). In step 1, as no single UE can occupy the whole PRBs in an integral number of subframes, we skip this step. In step 2, we construct a set \( \mathcal{X} = \{2, 4, 3, 6, 8\} \) based on the nil value of each UE. Given \( \lceil N(B_j)/4 \rceil = 3 \) and \( \lceil N(B_j)/2 \rceil = 5 \), we run LA(\( \mathcal{X}, N(B_j) \)) as follows: 1) \( s_1 = 4 + 6 + 8 = 18 > \tilde{N}(B_j) \), and the condition does not hold; 2) \( s_2 = 4 + 6 \leq \tilde{N}(B_j) \), so \( s = s_2 \) and \( \mathcal{L} = \{4, 6\} \); 3) since \( \tilde{N}(B_j) - s = 0 \), we return \( \mathcal{L} \). Thus, we first deploy PRBs for \( u_2 \) and \( u_4 \), followed by the PRBs of \( u_1 \) and \( u_3 \) (by step 2 again). Then, since \( \mathcal{U}_j \) has only \( u_2 \), we deploy \( u_3 \) PRBs for it. Theorem 2 shows that our PDMAS method has an approximation ratio of 3/4 to the optimal solution, and Lemma 4 analyzes the time complexity of the DRX setting scheme.

**Theorem 2.** The PDMAS method is a 3/4-approximation algorithm.

**Proof:** The work [33] proves that the LA(\( \mathcal{X}, t \)) algorithm is a 3/4-approximation algorithm to solve the subset-sum problem. In Theorem 1, we also verify that the subset-sum problem can be reduced to the PDMAS problem. As the PDMAS method adopts the LA(\( \mathcal{X}, t \)) algorithm to find the solution, it is a 3/4-approximation algorithm.

**Lemma 4.** Given \( \tilde{n}_U \) UEs, the DRX setting scheme spends at most \( O(n_U^2) \) time.

**Proof:** The DRX setting scheme uses the PDMAS method to arrange PRBs for UEs. In its step 1, we first check if the condition of \( n_i \mod N(B_j) = 0 \) holds for each UE. This step thus spends \( O(n_U) \) time. Then, step 2 adopts the LA(\( \mathcal{X}, t \)) algorithm to get the list \( \mathcal{L} \), which takes \( O(|\mathcal{X}|) \) time [33]. The worst case occurs when LA(\( \mathcal{X}, t \)) returns \( \mathcal{L} \) with only two items in each iteration of step 2. Thus, it spends time of \( O(n_U) \), \( O(n_U - 2) \), \( O(n_U - 4) \), \( \cdots \), and \( O(2) \) in the 1st, 2nd, 3rd, \( \cdots \), and \( \lceil N(B_j)/4 \rceil \)th iterations, respectively. Obviously, this is an arithmetic sequence. So, both steps 2 and 3 take time of

\[
\sum_{i=0}^{\lceil N(B_j)/4 \rceil} O(n_U - 2i) = O\left(\frac{n_U(n_U + 2)}{4}\right) = O(n_U^2).
\]

After deploying PRBs for each UE, we can get its \( T^{\text{ON}} \) and \( T^{\text{OFF}} \) values, which requires to search all UEs once. This operation takes \( O(n_U) \) time. Therefore, the time complexity of the DRX setting scheme is \( O(n_U) + O(n_U^2) + O(n_U) = O(n_U^2) \), which verifies the lemma.

**4.3 Discussion**

We give the rationale of our JIPM mechanism. The CQI evaluation scheme in stage 1 not only decides the lengths of ABS and DRX cycles, but also computes SINR of each UE with or without ABS. To meet QoS demands of different flows, we pick the strictest delay budget to set the long DRX cycle. Besides, to allow macrocell UEs going to sleep in ABSs, we also synchronize both DRX and ABS cycles. Based on SINR and cycle length, the resource estimation scheme in stage 2 calculates the number of PRBs used to send out GBR and non-GBR data of each UE in ABSs and non-ABSs.

The ABS setting scheme in stage 3 is the core of JIPM, which adjusts the ABS ratio and finds the actual number of PRBs given to UEs. Although the ABS method facilitates data transmissions in picocells by improving their signal quality, it degrades macrocell throughput. To improve network throughput while supporting fairness, the ABS setting scheme adopts three policies:

- **P1.** GBR flows always have the highest priority to meet their delay requirements.
- **P2.** Traffic loads of a macrocell and its picocells should be balanced.
- **P3.** In each cell, we increase its throughput but should avoid starving many UEs.

For policy P1, we first give PRBs to GBR flows. Only when an eNB has residual PRBs will they be allocated to non-GBR flows (by step 5). Thus, if PRBs are not enough, we can meet QoS demands of GBR flows as much as possible. For policy P2, we start from \( \gamma = 0 \) and iteratively increase \( \gamma \) by one to reduce loads of picocells, until the condition of Eq. (17) is satisfied
Finally, the DRX setting scheme in stage 4 decides both $T^{\text{ON}}$ and $T^{\text{OFF}}$ of each UE. To synchronize the sleeping period of macrocell UEs with ABSs, we align the deployment of PRBs with the ABS cycle, as Fig. 3 shows. Besides, we deploy non-ABS-based PRBs for picocell UEs, followed by ABS-based PRBs. In this way, we can also synchronize the ABS cycles of macrocell and picocell UEs. Even though different strategies of PRB deployment do not change the total number of PRBs used (referring to the examples in Fig. 3), they will affect the on-duration timer $T^{\text{ON}}$ of each UE, thereby deciding its energy consumption. Thus, we propose the NP-hard PDMAS problem to minimize $T^{\text{ON}}$ values of UEs, and develop a 3/4-approximation method. With these designs, the JIPM mechanism can integrate the ABS and DRX methods to jointly manage interference and power in LTE-A HetNets. Theorem 3 analyzes the time complexity of JIPM.

**Theorem 3.** Given $\bar{n}_{\text{UE}}$, $\bar{n}_E$, $\bar{n}_C$ in Eqs. (1)–(2), $\bar{m}_E$, $\bar{m}_C$ flows, and $(\bar{m}_C + P^{ABs})$, $\bar{m}_E$ $\bar{m}_F$ $\bar{m}_F$, the JIPM mechanism has time complexity of $O((\bar{n}_{\text{UE}}\bar{n}_E) \cdot (\bar{m}_C + P^{ABs}) + \bar{m}_E \log \bar{m}_F + \bar{m}_F^2)$ in the worst case.

Proof: JIPM comprises the four schemes in Sections 4.2.1–4.2.4. According to Lemmas 1–4, the worst-case time complexity of JIPM is $O((\bar{n}_{\text{UE}}\bar{n}_E) + 20(\bar{m}_F) + O(P^{ABs}) + \bar{m}_E \log \bar{m}_F + O(\bar{m}_F^2) = O((\bar{n}_{\text{UE}}\bar{n}_E) \cdot (\bar{m}_C + P^{ABs}) + \bar{m}_E \log \bar{m}_F + \bar{m}_F^2)$, which verifies the theorem.

### 5 Performance Evaluation

We evaluate the performance of our JIPM mechanism by LTE-Sim, which is an open-source simulation framework to simulate transmission behavior in LTE-A [34]. Below, we introduce standard models for DRX considered by 3GPP, followed by the settings of simulation. Then, we discuss the experimental results, including energy efficiency, network throughput, energy consumption of UEs, GBR packet dropping, and effect of UE distribution.

#### 5.1 Standard Models for DRX

There are two standard models used to describe UE behavior in DRX, namely 3-state and Nokia. The 3-state model is a semi-Markov process that contains the states of power active, light sleep, and deep sleep, as shown in Fig. 4(a). The power-active state comprises a sequence of adjacent active time intervals, which corresponds to the duration of one single packet call. A UE follows DRX short cycles when it stays in the light-sleep state. When a UE enters the deep-sleep state, it actually follows DRX short cycles when it stays in the light-sleep state. Therefore, we choose the Nokia model in our simulations. However, since JIPM does not consider the light-sleep state (i.e., short DRX cycle), a UE directly changes from the deep-sleep state to idle state, which spends $39 \text{ mW}$ to switch to the idle state.

#### 5.2 Simulation Settings

Table 3 gives simulation parameters of eNBs in LTE-Sim, where $\zeta$ is the distance between the eNB and a UE, which is measured in kilometers and meters for macrocells and picocells, respectively. Since the LTE-Sim program does not support the ABS method, we do two modifications. First, as LTE-Sim applies the small-cell scenario of femtocells, we increase the transmitted power of a femtocell eNB to 30 dBm to imitate a picocell scenario of femtocells, we increase the transmitted power macrocell: 46 dBm, picocell: 30 dBm cell range macrocell: 1 km, picocell: 100 m path loss [38] macrocell: 128.1 + 37.6 log $\zeta$ picocell: 38 + 30 log $\zeta$ shadowing fading log-normal distribution with 0 dB mean and 8 dB standard deviation penetration loss 10 dB fast fading Jakes model (for Rayleigh fading) CRE bias 9 dB
We first measure the amount of energy efficiency, which is defined by the ratio of network throughput to energy consumption. In general, higher energy efficiency implies that UEs spend less energy to achieve higher throughput (i.e., they can better utilize their energy to get data). Fig. 6 gives the simulation result by increasing the number of UEs from 90 to 360. NDC always has the lowest energy efficiency, as it does not allow UEs sleeping to save energy (even if they have nothing to receive). Both SLC and LLC use fixed DRX parameters, so they also result in lower energy efficiency, as they cannot change the sleeping and awake time of UEs. Since SLC uses short DRX cycles to let UEs receive more data, it can overtake LLC with more UEs.

Both DXD and JIPM adjust $T^{ON}$ and $T^{OFF}$ for each UE based on the network condition, so they achieve much higher energy efficiency than others. When the number of UEs grows, their energy efficiency decreases, as network throughput is saturated but more UEs spend energy on getting data. For DXD, a larger ABS ratio $\delta$ results in lower energy efficiency. The reason is that macrocell throughput greatly reduces in DXD when $\delta$ grows. JIPM integrates the ABS and DRX methods, so it can adaptively co-adjust $\delta$ and DRX parameters. Thus, JIPM always has higher energy efficiency than DXD, and the gap enlarges when $\delta$ increases.

### 5.4 Network Throughput

We then study network throughput in Fig. 7. In general, the more the UEs, the higher the throughput. Except for JIPM, network throughput greatly decreases in all schemes when $\delta$ grows, because there are more ABSs to let macrocell eNBs stop sending data. From Fig. 7, NDC has higher throughput, followed by DXD, SLC, and LLC. The reason is that NDC makes UEs always active to get data while DXD adjusts DRX parameters based on CQI, so they achieve higher throughput than SLC and LLC. By jointly managing cell interference and UE power, JIPM has the highest throughput among all schemes in most cases.

We also evaluate the percentage of macrocell throughput to total throughput by each scheme, as Fig. 8 shows. For NDC, SLC, LLC, and DXD, the percentage obviously goes down as $\delta$ increases, since macrocell UEs get fewer data with a growing number of ABSs. When $\delta = 0.9$, both SLC and LLC have little macrocell throughput. In NDC and DXD, macrocells contribute less than 10% of total throughput with more than 180 UEs. The result shows that these schemes starve macrocell UEs with a large ABS ratio, even if around 55% of UEs are served by macrocells. On the contrary, JIPM uses Eq. (17) to balance cell loads, so its percentage can keep higher than 40%.

### 5.5 Energy Consumption of UEs

Fig. 9 gives the amount of energy spent by UEs in different schemes. Obviously, NDC lets UEs consume much more energy than other schemes that use the DRX method, which shows the superiority of DRX in energy conservation. On the other hand, SLC makes UEs spend more energy than LLC, DXD, and JIPM. The reason is that SLC uses fixed short DRX cycles, so UEs have to wake up (and spend energy accordingly) more frequently. This experiment verifies that JIPM can achieve lower energy consumption of UEs as well as both LLC and DXD.

Recall that we give an intuitive method to deploy the PRBs of UEs based on their sequence in Section 4.2.4 (referring to Fig. 3(a)). We also compare the amount of energy consumption by this intuitive method (denoted by “JIPM-SEQ”) and the PDMAS method (denoted by “JIPM-PDMAS”). Fig. 10 presents the experimental result. Obviously, when there are...
more UEs, the PDMAS method can save more energy than the intuitive method. The reason is that the PDMAS method can find out more UEs such that their sum of allocated PRBs is divisible by $\hat{N}(B_j)$ and thus shrink their aware time. In particular, the PDMAS method further saves 1.71% to 4.39% energy of UEs than the intuitive method, when the number of UEs increases from 90 to 360.

5.6 GBR Packet Dropping

Fig. 11 shows the dropping ratio of GBR packets. Intuitively, the dropping ratio can reduce by increasing throughput. From Fig. 7, NDC has higher throughput, followed by DXD, SLC, and LLC. Thus, LLC results in the highest dropping ratio, followed by SLC, DXD, and NDC. Besides, these schemes have higher dropping ratios with a larger ABS ratio. Comparing
results demonstrate that JIPM performs well under different conditions. These schemes have the best energy efficiency in each HetNet. Moreover, JIPM minimizes the impact on the macrocell, all schemes result in the lowest dropping ratio. This experiment verifies that JIPM can better support QoS for GBR applications.

5.7 Effect of UE Distribution
Finally, we investigate how different UE distributions affect performance of NDC, SLC, LLC, DXD, and JIPM, where \( \delta \) is set to 0.5. Fig. 12 gives the amount of energy efficiency in the three basic HetNets, and Fig. 13 shows their dropping ratios of GBR packets. Recall that there are 5/6, 1/3, and 1/2 of UEs served by the macrocell in HetNets A, B, and C, respectively. Since the ABS method benefits picocells by reducing the transmission opportunity of the macrocell, all schemes result in the worst performance in HetNet A. From Fig. 12, JIPM always has the best energy efficiency in each HetNet. Moreover, it results in a lower dropping ratio than others in Fig. 13. These results demonstrate that JIPM performs well under different distributions of UEs, which verifies the effectiveness of its joint management of interference and power.

6 Conclusion and Future Work
LTE-A uses the ABS and DRX methods to reduce signal interference and save energy of UEs, respectively. Existing studies aim to either find the ABS ratio or configure the DRX parameters. None of them considers integrating these two methods, even though they share some similarities. Therefore, this paper develops a four-stage JIPM mechanism to co-adjust the parameters of ABS and DRX, so as to improve LTE-A performance. JIPM finds CQIs of each UE with or without ABS, and estimates the amount of resources required by the UE. It computes the ABS ratio based on the principles of raising picocell GBR throughput and avoiding starving macrocell UEs. Then, JIPM decides both \( T_{ON}^i/C \) and \( T_{OFF}^i/C \) values of a UE to shorten its wake-up time for energy conservation. Through simulations by LTE-Sim, we show that JIPM greatly improves energy efficiency, which means that it allows UEs receiving more data by spending less energy. Comparing with other schemes, JIPM also results in a much lower dropping ratio of GBR packets, which verifies that it can better support QoS for real-time traffics.

However, JIPM addresses only DRX long cycles for the purpose of energy saving. In the future work, we will consider adjusting the parameters of short cycles (e.g., \( P_{SFC} \)) to provide fine-tuning of the DRX method. Besides, JIPM first decides the ABS ratio followed by DRX parameters. It deserves further investigation on how to feed back the result of DRX configuration to the decision of the ABS ratio, so as to achieve tighter combination of both ABS and DRX methods.

References


