

# A Distributed Computation Offloading Strategy in Small-Cell Networks Integrated With Mobile Edge Computing

Lichao Yang , Heli Zhang , Xi Li , Hong Ji, Victor C. M. Leung

一. 问题: most of them ignore considering the interests interaction between small cells and MDs.

二. 系统模型:

## A. Small Cell Network Integrated With MEC Model

is composed of multiple devices and multiple Micro base stations (MBSs), the MBSs are connected to the core network through wired optical fibers. the MBSs  $\Rightarrow N = \{1, \dots, N\}$

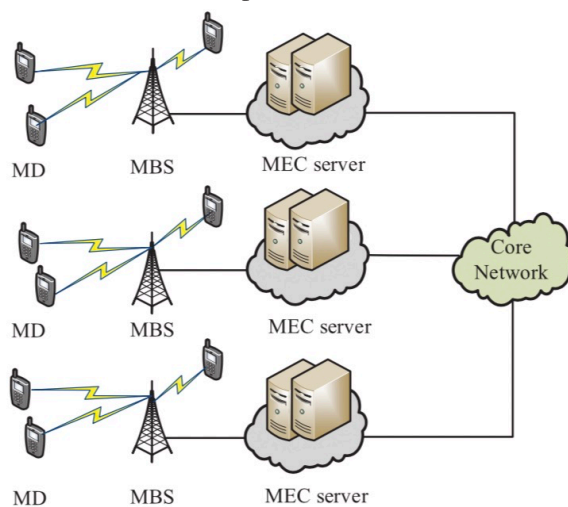


Fig. 1. An illustration of small cell network integrated with MEC.

MEC-MBSs: 具有MEC服务器的MBS, MBS安装有MEC服务器, MBS具有MEC功能, 能够在SCN边缘提供增强的计算能力

$I = \{1, \dots, I\}$  : the set of MDs , 每个MD仅通过MBS与一个MEC服务器关联

信道接入: multi-device 和 multi-MEC 正交频分多址 (OFDMA) , 其中每个MEC-MBS中的频谱分为  $M=1, 2, \dots, M$  信道, 并且信道在一个MEC-MBS中是正交的。

卸载的开销最小化问题将共同优化通信模型和计算模型中的通信和计算资源分配。

## B. Communication Model

$$a_{i,n}^m \in \{0, 1\}$$

作为MD i的通道和MEC服务器的决策集,  $=0$ 表示MD i选择不通过信道m访问MEC-MBS n,  $=1$ 表示MD i通过信道M访问MEC-MBS n。

MD i 决定在MD本地计算其任务, 假设MD i通过信道m访问MEC-MBS n, 数据传输速率为:

$$r_{i,n}^m = W \log_2 \left( 1 + \frac{p_{i,n} g_{i,n}^m}{\sigma^2 + K_{i,n}^m} \right)$$

信道带宽
MD i的传输功率
MD i和MEC-MBS n之间通过信道m链路的信道增益
高斯噪声
表示卸载时选择同一信道的其他MDs对m信道的干扰

## C. Computation Model

$$L_i = \{b_i, w_i, t_i^{max}\}$$

MD i的计算任务
总计算能力
计算任务的大小
MD i可以容忍的最大延迟

$\eta_i$ 表示任务中每MB所需的CPU周期

$$\eta_i = w_i / b_i$$

### 1) Local Computing:

本地计算的开销包括两部分：计算任务的本地执行延迟 $t$ 和能耗 $e$ 。

$$t_i^l = w_i / f_i^l \quad E_i^l = w_i \varepsilon_i^l$$

MD i的计算能力 (即每秒CPU周期数)
MD i每CPU周期能耗系数

所以本地计算开销为：

$$Z_i^l = \lambda_1 t_i^l + \lambda_2 E_i^l$$

$$0 \leq \lambda_1, \lambda_2 \leq 1 \quad (\lambda_1 + \lambda_2 = 1)$$

为了反映不同研究场景的特点，设置不同的加权参数，以反映不同研究场景的总体需求

### 2) Edge Computing:

边缘计算的开销主要来自计算任务的传输过程和计算任务的计算过程。

边缘计算的总延迟 = 计算任务传输的延迟 + MEC-MBS上计算任务计算的延迟

$$t_{i,n}^{m,s} = b_i / r_{i,n}^m \quad t_{i,n}^x = w_i / f_{i,n}^x$$

MD i在边缘计算中的总延迟为：
表示分配给MD i的MEC-MBS N的计算能力

MD i在边缘计算中的总延迟为：

$$t_{i,n}^{m,e} = t_{i,n}^{m,s} + t_{i,n}^x$$

通道m从MD i到MEC-MBS n的能量消耗 = 传输能耗 + 执行能耗

$$E_{i,n}^{m,s} = p_{i,n} b_i / r_{i,n}^m \quad E_{i,n}^x = w_i \varepsilon_n$$

$\varepsilon$ 是每个MEC服务器实现CPU周期的能源成本

通道m从MD i到MEC-MBS n的能量消耗为：

$$E_{i,n}^{m,e} = E_{i,n}^{m,s} + E_{i,n}^x$$

所以边缘计算总开销为：

$$Z_i^e = \lambda_1 \sum_n \sum_m a_{i,n}^m t_{i,n}^{m,e} + \lambda_2 \sum_n \sum_m a_{i,n}^m E_{i,n}^{m,e}$$

忽略了下行延迟

## 三. 问题定义：

**优化目标：**最小化开销。分布式开销最小化问题，旨在共同优化每个MD的能耗和延迟；

问题表示为：

$$\min_{a_i \in \{0,1\}} Z_i(a_i, a_{-i}), \quad \forall i \in \mathbb{I}$$

$a_i$  : MD i的计算模式选择,  $a_{-i}$  : 所有其他MDs (MD i除外) 选择的计算模式

$\{Z_i\}_{i \in \mathbb{I}}$  : 每个MD i的开销函数

$$Z_i(a_i, a_{-i}) = \begin{cases} Z_i^l(1 - a), & \text{if } a_i = 0. \\ Z_i^e(a), & \text{if } a_i = 1. \end{cases}$$

约束:

1) 负载约束—分配给MEC-MBS  $n$ 上执行的计算任务的全部计算资源不超过MEC-MBS  $n$ 的计算资源 $c_n$

$$\sum_{i \in \mathbb{I}} \sum_{m \in \mathbb{M}} a_{i,n}^m f_{i,n}^x \leq c_n, n \in \mathbb{N}.$$

2) 延迟约束— 阈值参数 $\kappa_1$ 来判断MD  $i$ 何时可以通过边缘计算来执行计算密集型任务

$$t_{i,n}^{m,e} < \kappa_1 t_i^l \quad (0 < \kappa_1 \leq 1)$$

3) Interference constraint — MD测量的总干扰噪声功率应小于MD  $i$ 的信号功率,  $\kappa_2$  表示阈值

$$(\sigma^2 + \sum_{l=1, l \neq n}^N \sum_{j=1, j \neq i}^I a_{j,l}^m p_{j,l} g_{j,n}^m) / p_{i,n} g_{i,n}^m < \kappa_2 (0 < \kappa_2 \leq 1)$$

4) 能耗约束— 如选择边缘计算, 边缘计算的能耗不应高于本地计算的能耗,  $\kappa_3$  表示阈值

$$E_{i,n}^{m,e} < \kappa_3 E_i^l \quad (0 < \kappa_3 \leq 1)$$

#### 四. 算法: POTENTIAL GAME BASED OFFLOADING SCHEM

1) 将上面的开销最小化问题描述为一个策略博弈模型

$$\Gamma = (\mathbb{I}, \{A_i\}_{i \in \mathbb{I}}, \{Z_i\}_{i \in \mathbb{I}})$$

2) 引入一个 potential function 证明所表述的博弈是一个 potential game

$$P(a) = \frac{1}{2} \sum_{n=1}^N \sum_{i=1}^I \sum_{l \neq n} \sum_{j \neq i} C_{i,n}^m C_{j,l}^m I_{\{a_i=a_j:m\}} I_{\{a_i=1\}} \\ + \sum_{n=1}^N \sum_{i=1}^I C_{i,n}^m C I_{\{a_i=0\}}$$

其中定义 $C$ 作为计算卸载的阈值

$C = \beta_1 \kappa_1 + \beta_2 \kappa_2 + \beta_3 \kappa_3$ , 其中 $\beta_j$  ( $j=1, 2, 3$ ) 分别为潜伏期、干扰和能耗的加权参数

$$C_{i,n}^m = \beta_1 \left( \frac{t_{i,n}^{m,e}}{t_i^l} \right) + \beta_2 \left( \frac{\sigma^2 + \sum_{l=1, l \neq n}^N \sum_{j=1, j \neq i}^I a_{j,l}^m p_{j,l} g_{j,n}^m}{p_{i,n} g_{i,n}^m} \right) \\ + \beta_3 \left( \frac{E_{i,n}^{m,e}}{E_i^l} \right) \quad (10) \quad a_{i,n}^m = \begin{cases} 1, & \text{if } C_{i,n}^m < C \\ 0, & \text{if } C_{i,n}^m > C. \end{cases}$$

3) 分析多设备和多MEC之间基于游戏的有效卸载方案

思想: 当MDs想要更新他们的决策时, 他们将向所有MEC-MBSs广播他们的参数。然后, 在每次迭代中, PGOA允许花费最少开销的卸载MD更新其决策。经过多次迭代, 所有MDS的卸载决策将进入一个稳定的状态, 称为开销最小化问题的次最优解。

PGOA的操作流程:

1) MDS在决策槽中随机初始化卸载决策

2) 在一个给定的决策时隙 $\tau$ , 根据干扰和上行链路数据率计算当前的最优卸载决策 $a$ , 得出较好的响应集 $\theta(\tau)$ 。如果MD  $i$ 的决策集 $\theta(\tau)$ 满足不为空的要求, MD  $i$ 将向服务器发送RTU消息, 争取UD机会。

3) UD竞争过程采用基于随机退避的机制来设置两个广播之间的时间 $\tau$ 作为决策时段。假设MD  $i$ 在给定时间段 $\tau$ 收到更新决策的信息, 以便在卸载时将开销最小化, MD将从最佳响应集更新自己的决策。然后, MD将向所有其他MDs广播该消息, 以表明它赢得了UD机会。没有机会更新他们的决策的MDs, 将保留最初的卸载决策 $a_i(\tau+1) = a_i(\tau)$ 。经过有限次迭代, 所有MDS都达到了相互的纳什平衡状态。

---

**Algorithm 1** Potential Game Based Offloading Algorithm

---

**Initialization:**

The strategy set of MD  $i$ :  $a_i \in \{0, 1\}$ ;

Set initial decision slot:  $\tau = 0$ ;

The number of decision slot:  $\Xi$ ;

Each MD  $i$ 's initial computation offloading decision  $a_i(\tau) = 0$ ;

2: **for** each decisions slot  $\tau \leq \Xi$  **do**

    compute the current wireless channel interference and the corresponding uplink data rate; 计算当前无线信道干扰和相应的上行数据速率

4: **if** MD  $i$  is in decision slot  $\tau$  **then**

compute the threshold  $C_{i,n}^m$  of each MD  $i$  in next slot  $\tau$  and make a preliminary judgment;

6: compute the better response  $\Theta_i(\tau)$  of each MD  $i$ ;

**end if**

8: **while**  $\Theta_i(\tau) \neq \emptyset$  **do**

    each MD in  $\Theta_i(\tau)$  send RTU message to servers and contend for UD opportunity;

10: **if** MD  $i$  wins the UD opportunity **then**

    compute the best response of MEC  $i$ :  $\Theta_i^*(\tau) \in \Theta_i(\tau)$ ;

12: update  $a_i$  in  $\Theta_i(\tau)$ , i.e.,  $a_i(\tau + 1) = \Theta_i^*(\tau)$  ;

**else**

14: keep the offloading decision  $a_i$  unchanged:  $a_i(\tau + 1) = a_i(\tau)$ ;

**end if**

16: for MDs  $j$  not wins the UD opportunities, they will keep decisions unchanged as well;

**end while**

18: **end for**

直到MEC系统中没有rtu信息

until no RTU message in the MEC system.

---

## 五. 实验:

- 1) 不同的 $\lambda$ 参数, MD数量和系统开销的关系
- 2) 阈值C的不同参数, 迭代次数和系统开销的关系。
- 3) 本地计算算法与PGOA算法, 系统开销与计算的不同数据大小的关系
- 4) 不同算法, 系统延迟与不同数量的移动设备的关系
- 5) 不同算法, 系统开销与不同数量的移动设备的关系

small-cell networks (SCNs)

Nash Equilibrium (NE).

the efficiency ratio (ER)

Micro base stations (MBSs)

request-to-update (RTU)

the update decision (UD)