Computation Offloading and Resource Allocation in Wireless Cellular Networks with Mobile Edge Computing

Chenmeng Wang, Chengchao Liang, F. Richard Yu, Qianbin Chen, Lun Tang

1.问题的出现：传统的无线蜂窝网络不能满足高数据率和高计算能力的成倍增长的需求。

2.解决方案：

环境：wireless cellular networks 无线蜂窝网络

移动边缘计算：

计算卸载：提高计算能力，减少延迟，减少成本

网络内缓存：减少重复的信息传输-> 减少延迟和增加吞吐量

3.框架：计算卸载、资源分配和内容缓存的联合设计

4.系统模型：

A. Network Model 网络模型

异质网络结构，在一个 macro cell 中部署多个低功耗、局部覆盖增强的small cells 、相同的无线资源可以在小蜂窝和宏蜂窝之间共享

The macro cell 通过蜂窝通信系统的核心网连接到因特网

MEC位于 the macro eNodeB (MeNB)中，所有N个small cell eNodeBs (SeNBs)连接到MeNB以及MEC服务器。

SeNBs有线连接到MeNB

single-antenna UEs and SeNBs

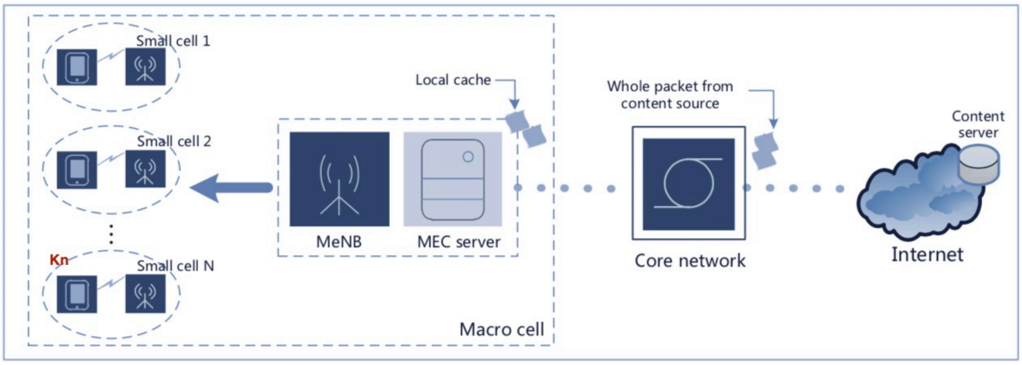

Fig. 1: Network model.

假设每个UE都有一个计算密集型和延迟敏感的任务要完成，每个UE可以通过与其关联的SeNB将计算卸载到MEC服务器，或者在本地执行计算任务。UEs从Internet请求内容，然后Internet内容将通过MeNB传输到UEs。

运营商情况：

移动网络运营商（MNO）拥有并运营无线网络的无线资源和物理基础设施，包括频谱、回程、无线接入网络、传输网络、核心网络等，

MEC系统运营商（MSO）拥有MEC服务器，从MNO租赁物理资源（如频谱和回程），并提供向UES提供移动边缘计算服务。MSO将向UES收取接收移动边缘计算服务的费用。

B. Communication Model 通信模型

akn ∈ {0,1}表示计算卸载决策，akn =0不卸载本地执行，akn =1卸载

small cells 间之间存在干扰。small cell 内UE相互干扰。干扰是从一个UE到邻近的SeNB

According to Shannon bound, UE kn 频谱效率：

传输功率密度　UE kn和SeNB n之间的信道增益

$$e_{k_n} = \log_2\left(1 + \frac{p_{k_n} G_{k_n,n}}{\sigma + \sum_{m=1, m\neq n}^{N} \sum_{i=1}^{K_m} p_{i_m} G_{i_m,n}}\right), \quad \forall n, k, \tag{1}$$

加性高斯白噪声的功率谱密度　UE im和SeNB n之间的信道增益

预期瞬时数据速率

Skn 表示为 small cell n分配给UE kn的无线电频谱百分比

$$R_{k_n}(\boldsymbol{a}, \boldsymbol{s}) = a_{k_n} s_{k_n} B e_{k_n}, \quad \forall n, k.$$

B表示整个可用频谱带宽

MeNB和MEC服务器之间的回程容量为L bps，SeNB n的回程容量为Ln bps。

限制：数据速率不能超过senb n的回程容量

$$\sum_{k_n \in \mathcal{K}_n} R_{k_n} \leq L_n$$

所有UE的总数据速率不能超过MeNB的回程容量

$$\sum_{n \in \mathcal{N}} \sum_{k_n \in \mathcal{K}_n} R_{k_n} \leq L$$

C. Computation Model 计算模型

每个UE 有一个计算任务， 其可以本地计算，也可以通过计算卸载在MEC服务器上远程计算

$$W_{k_n} \triangleq (Z_{k_n}, D_{k_n}),$$

Zkn 表示输入数据的大小，包括程序代码和输入参数

Dkn 表示完成计算任务Wkn所需的CPU周期总数

本地和MEC计算方法处理时间方面的计算开销：

1) Local Computing 计算开销：

计算执行时间

$$T_{k_n}^{(l)} = \frac{D_{k_n}}{f_{k_n}^{(l)}}$$

kn的计算能力，即每秒CPU周期数

2) MEC Server Computing 计算开销：

根据通信模型，发送计算输入数据的时间成本Zkn大小计算如下：

$$T_{k_n,off}^{(e)}(\boldsymbol{a}, \boldsymbol{s}) = \frac{Z_{k_n}}{R_{k_n}(\boldsymbol{a}, \boldsymbol{s})}$$

预期瞬时数据速率

MEC服务器在任务wkn上的执行时间如下

MEC服务器执行任务Wkn的执行时间

$$T_{k_n,exe}^{(e)} = \frac{D_{k_n}}{f_{k_n}^{(e)}}.$$

分配给UE kn的MEC服务器的计算能力

所以ue kn任务的总执行时间

$$T_{k_n}^{(e)}(\boldsymbol{a}, \boldsymbol{s}) = T_{k_n,off}^{(e)}(\boldsymbol{a}, \boldsymbol{s}) + T_{k_n,exe}^{(e)}$$

忽略了从MEC服务器到Ue-kn的计算结果传输的时间消耗。

D. Caching Model 缓存模型

hkn ∈ {0,1} 缓存策略。 hkn =1 MEC服务器缓存UE kn请求的内容 hkn =0不缓存

缓存降低macro cell 与Internet 之间的回程带宽

缓存UE kn请求的内容的奖励（减轻回程带宽）可以表示为：

系统中的平均单个UE的数据速率

$$Caching\ reward = q_{k_n} \bar{R} h_{k_n}$$

kn首先请求的内容的请求率（由其他UEs提供）

如请求的内容大小不变，请求率服从Zipf 普遍分布，则

$$q(i) = 1/i^{\beta}$$

i代表第i个最流行的内容，β是一个常量，其典型值为0.56
已知UE-kn首次请求的内容的大小，则其他Ue对相同内容的请求率可由上述方程式得出。

所有缓存内容的总和大小不能超过MEC服务器的总存储能力，所以

$$\sum_{n \in \mathcal{N}} \sum_{k_n \in \mathcal{K}_n} h_{k_n} o_{k_n} \leq Y$$

Y是MEC服务器的总存储能力，Okn是由UE kn首先请求的内容的大小。假设Okn为常数，Okn=1。

5.问题定义：

优化目标：最大化MSO的收益

收入：

（1）MSO将无线电资源分配给UE KN的净收入为

收取的费用单价（θn/bps）　租用频谱的单价（δn/hz）

$$\iota_{k_n} = s_{k_n} \Psi_{k_n} = s_{k_n} (\theta_n B e_{k_n} - \delta_n B - \eta_n B e_{k_n})$$

small cell n与macro cell之间回程的单价（ηn/bps）

（2）计算资源分配给UE-kn的净收入为：

$$\Omega_{k_n} = \lambda_n (c_{k_n} F / D_{k_n} - f_{k_n}^{(l)} / D_{k_n})$$

ckn 分配给UE kn的MEC服务器计算资源的百分比。

（3）UE kn首次请求缓存互联网内容的长期收益为：

$$\Lambda_{k_n} = \zeta q_{k_n} \bar{R} - \varpi$$

所以MSO的效用函数为：

$$U = \sum_{n \in \mathcal{N}} \sum_{k_n \in \mathcal{K}_n} u \left(a_{k_n} \iota_{k_n} + a_{k_n} \Omega_{k_n} \right) + h_{k_n} \Lambda_{k_n}$$

优化问题：　Let $\lambda_n F / D_{k_n} = \Phi_{k_n}$

$$\begin{aligned}
\underset{\boldsymbol{a},\boldsymbol{s},\boldsymbol{c},\boldsymbol{h}}{\text{Maximize}} \quad & \sum_{n \in \mathcal{N}} \sum_{k_n \in \mathcal{K}_n} a_{k_n} u \left(s_{k_n} \Psi_{k_n} + c_{k_n} \Phi_{k_n} \right) + h_{k_n} \Lambda_{k_n} \\
s.t. \quad C1: \quad & \sum_{k_n \in \mathcal{K}_n} a_{k_n} s_{k_n} \leq 1, \forall n \\
C2: \quad & \sum_{k_n \in \mathcal{K}_n} a_{k_n} s_{k_n} B e_{k_n} \leq L_n, \forall n \\
C3: \quad & \sum_{m \in \mathcal{N}/\{n\}} \sum_{k_m \in \mathcal{K}_m} a_{k_m} p_{k_m} G_{k_m,n} \leq I_n, \forall n \\
C4: \quad & \sum_{n \in \mathcal{N}} \sum_{k_n \in \mathcal{K}_n} a_{k_n} c_{k_n} \leq 1 \\
C5: \quad & a_{k_n} \left(\frac{c_{k_n} F}{D_{k_n}} - \frac{f_{k_n}^{(l)}}{D_{k_n}} \right) \geq 0, \forall k, n \\
C6: \quad & \sum_{n \in \mathcal{N}} \sum_{k_n \in \mathcal{K}_n} h_{k_n} \leq Y.
\end{aligned}$$

C1：频谱约束
C2：回程容量约束
C3：干扰约束
C4：计算资源总和约束
C5：每个UE kn的计算资源约束
C6：存储约束

6.问题转换：

上述优化问题是一个混合的离散和非凸优化问题，这些问题通常被认为是NP-hard问题

问题的转换由以下两个步骤组成：

1）二元变量松弛

2）乘积项的代换

转化为凸问题为：

$$
\begin{aligned}
\underset{\boldsymbol{a},\tilde{\boldsymbol{s}},\tilde{\boldsymbol{c}},\boldsymbol{h}}{\text{Maximize}} \quad & \sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} a_{k_n} u\left(\frac{\tilde{s}_{k_n}\Psi_{k_n}+\tilde{c}_{k_n}\Phi_{k_n}}{a_{k_n}}\right)+h_{k_n}\Lambda_{k_n} \\
s.t. \quad C1: \quad & \sum_{k_n\in\mathcal{K}_n} \tilde{s}_{k_n} \le 1, \forall n \\
C2: \quad & \sum_{k_n\in\mathcal{K}_n} \tilde{s}_{k_n} B e_{k_n} \le L_n, \forall n \\
C3: \quad & \sum_{m\in\mathcal{N}/\{n\}}\sum_{k_m\in\mathcal{K}_m} a_{k_m} p_{k_m} G_{k_m,n} \le I_n, \forall n \\
C4: \quad & \sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} \tilde{c}_{k_n} \le 1 \\
C5: \quad & \tilde{c}_{k_n}\frac{F}{D_{k_n}} - a_{k_n}\frac{f_{k_n}^{(l)}}{D_{k_n}} \ge 0, \forall k, n \\
C6: \quad & \sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} h_{k_n} \le Y \\
C7: \quad & a_{k_n} \ge \tilde{s}_{k_n}, a_{k_n} \ge \tilde{c}_{k_n}, \forall k, n.
\end{aligned}
$$

7.问题分解：

$$
U'' = \sum_{n\in\mathcal{N}}\sum_{k_n\in\mathcal{K}_n} \hat{a}_{k_n}^n u\left(\frac{\tilde{s}_{k_n}\Psi_{k_n}+\hat{c}_{k_n}^n\Phi_{k_n}}{\hat{a}_{k_n}^n}\right)+\hat{h}_{k_n}^n\Lambda_{k_n}
$$

8.算法 — ADMM的算法

没理解

9.实验：

对比：ADMM算法和集中式算法

维度：

随着小单元总数的增加，卸载所有UE的百分比变化

系统中所有UE的频谱和计算资源分布

总减少回程使用量

MSO（效用值）相对于计算能力增加的收入

MSO（效用值）相对于越来越多的小单元的收入

执行计算任务的系统中的平均UE时间消耗