A Low-Cost Edge Server Placement Strategy in Wireless Metropolitan Area Networks

Yongzheng Ren*, Feng Zeng*, Wenjia Li†, Lin Meng‡

一. 问题的提出：

之前的关于边缘服务器防止的研究都是为了最小化用户和边缘服务器之间的平均访问延迟，没有考虑边缘服务器供应商的成本。部署边缘服务器的成本主要与两个因素有关：场地租金和计算需求。

场地租金：选择越多的位置来部署边缘服务器，成本就越高。

计算需求：计算需求越大，边缘服务器的数量就越多，成本越高。边缘服务器的维护和管理成本如人工成本将随着边缘服务器数量的增加而增加。

在何处放置边缘服务器以及如何放置边缘服务器来最小化边缘服务器提供商的成本。

—— 边缘服务器布局问题

二. 目标：在保证接入延迟等QoS约束的同时最小化边缘服务器数量

三. 挑战：一是如何选择与边缘服务器共处的最少数量的AP来为所有用户提供良好的服务，二是如何选择边缘服务器来执行用户的任务。

四. 解决方案：

4.1 架构：

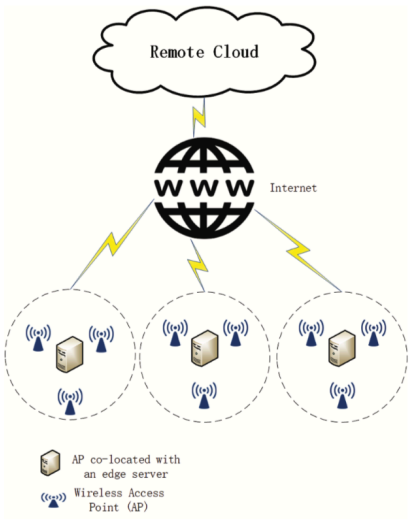


Fig. 1. Mobile Edge Computing Architecture.

移动边缘计算的体系结构由远程云、互联网、AP和边缘服务器组成。将WMAN划分为不相交的集群，这些集群包含不同数量的AP，在集群中部署边缘服务器。对于需要卸载任何具有计算任务的AP，可以选择集群内的边缘服务器，并且边缘服务器通过因特网连接，也可以选择远程云进行任务卸载。

4.2 建模：WMAN环境下

给定一个连通的无向图G=(V,E)。存在边(u，v)∈E，u∈V，v∈V当且仅当AP u和AP v通过通信链路连接。将n=|V|表示为AP的数量，将m=|E|表示为AP之间的链接的数量。用d(u，v)表示节点u和节点v之间的距离。用的邻接矩阵来表示图G=(V,E)

4.3 问题的定义：

给定WMAN G = (V, E) 和延迟上界Dmax 。目标找到最小子集K ⊂ V 使得可以在Dmax 限制内将V中的所有APs连接起来

xi表示是否在APi中部署边缘服务器（xi=1部署了，xi=0-没部署）

y(i,j) , j ∈ K ,表示AP i的工作负载是否分配给边缘服务器j （y(i,j)=1 分配了 y(i,j)=0 未分配）

c(j) 表示边缘服务器j的容量

r(i) 表示AP i的计算需求

D(u, v) 表示AP u和AP v之间的延迟

目标函数：—— 可以表示为整数线性规划（ILP）

$$min \sum_{i \in V} x_i$$

约束：

$$\sum_{i \in V} y_{(i,j)} \cdot r(i) \leq c(j), \forall j \in K \quad (1)$$

（1）确保每个边缘服务器都有足够的容量来处理分配给它的所有计算任务

$$\sum_{j \in K} y_{(i,j)} \cdot d(i,j) \leq d_{max}, \forall i \in V \quad (2)$$

（2）确保每个AP与相应边缘服务器之间的距离不超过距离上限，以满足延迟的要求。

$$\sum_{j \in K} y_{(i,j)} = 1, \forall i \in V \quad (3)$$

（3）表示每个AP只能分配给一个边缘服务器

$$x_i \in \{0,1\}, \forall i \in V \quad (4)$$

$$y_{i,j} \in \{0,1\}, \forall i \in V, j \in K \quad (5)$$

$$c(j)>0, \forall j \in K \quad (6)$$

$$r(i) \geq 0, \forall i \in V \quad (7)$$

五. 解决算法：基于贪婪的最小支配集算法(Greedy-based Extended Dominating Set Algorithm)

整数线性规划（ILP）问题 —— 转化为图论中的最小支配集问题

A. One Hop Constraint

在dmax=1的情况下，边缘服务器的放置问题可以转化为计算给定图的最小控制集的问题。

图G=（V，E）的支配集是V的子集D，这样不在D中的每个顶点都至少与D的一个成员相邻

B. Multi-Hop Constraint

图G=（V，E）的一个i层控制集是v的一个子集di，这样对于不在di中的每个顶点，在i跃点中顶点和di的至少一个成员之间存在连接。

使用贪心方法计算图G的最小扩展控制集di，选择一个extdeg值(extdeg（v，i）定义为d（v，u）不超过i的节点数)最大的节点作为每个while循环中未覆盖结点u的边缘服务器部署位置。

六. 实验：

6.1. 随机拓扑生成：

需要生成随机拓扑来为我们的实验建模真正的WMAN。γ表示所有AP的覆盖范围，δ代表两个APS之间的最小距离。给定Mkm×Mkm的随机区域，AP数N，γ和δ，我们逐步生成随机拓扑。

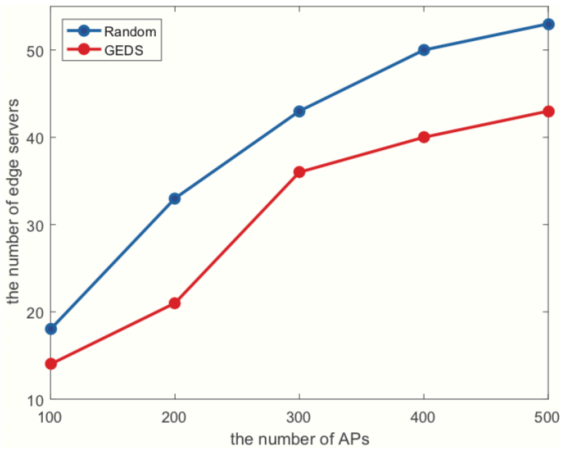
添加αp的条件：一个条件是AP必须覆盖现有AP，另一个条件是它们之间的距离不能超过γ。

直到添加n个

6.2. 性能评估 -- 根据所需的边缘服务器数量来评估GEDS的性能。

1. 通过改变网络大小来评估GEDS算法相对于随机算法的性能。

实验条件：在30km×30km的范围内，延迟约束为2，γ=2，δ=1，我们将APS的数量从100改为500



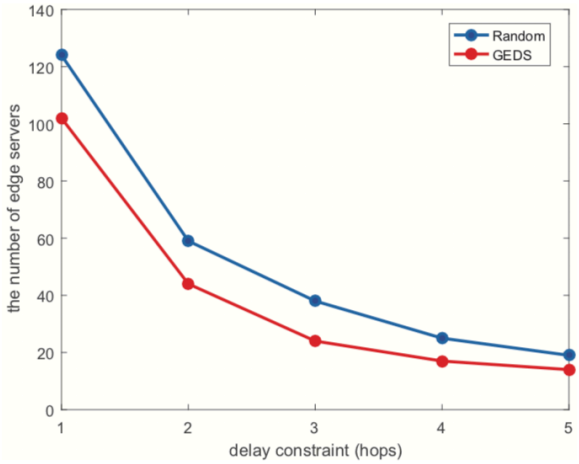
x坐标：APs的数量

y坐标：边缘服务器的数量

边缘服务器的数量随着接入点数量的增加而增加。随着网络规模的扩大，需要添加的额外边缘服务器数量减少

2. 延迟约束来评估GEDS算法的性能。

实验条件：网络面积仍为30km×30km，接入点数量为100个, γ=2，δ=1。



x坐标：延迟约束

y坐标：边缘服务器的数量

随着延迟约束的增加，APS的数量减少。这意味着用户可以承受的时间越长，我们需要的边缘服务器就越少。