

基于云雾混合计算的车联网联合资源分配算法

唐伦 肖娇* 魏延南 赵国繁 陈前斌

(重庆邮电大学通信与信息工程学院 重庆 400065)

(重庆邮电大学移动通信技术重点实验室 重庆 400065)

摘要: 针对车联网业务的低时延、低功耗需求及海量设备计算卸载引起的网络拥塞问题, 该文提出一种在云雾混合网络架构下的联合计算卸载、计算资源和无线资源分配算法(JODRAA)。首先, 该算法考虑将云计算与雾计算结合, 以最大时延作为约束, 建立最小化系统能耗和资源成本的资源优化模型。其次, 将原问题转化为标准二次约束二次规划(QCQP)问题, 并设计一种低复杂度的联合卸载决策和计算资源分配算法。进一步, 针对海量设备计算卸载引起的网络拥塞问题, 建立卸载用户接入请求队列的上溢概率估计模型, 提出一种基于在线测量的雾节点时频资源配置算法。最后, 借助分式规划理论和拉格朗日对偶分解方法得到迭代的带宽和功率分配策略。仿真结果表明, 该文算法可以在满足时延需求的前提下, 最小化系统能耗和资源成本。

关键词: 车联网; 雾计算; 计算卸载; 资源分配

中图分类号: TN929.5

文献标识码: A

文章编号: 1009-5896(2020)08-1926-08

DOI: 10.11999/JEIT190306

Joint Resource Allocation Algorithms Based on Mixed Cloud/Fog Computing in Vehicular Network

TANG Lun XIAO Jiao WEI Yannan ZHAO Guofan CHEN Qianbin

(School of Communication and Information Engineering, Chongqing University of Post and Telecommunications, Chongqing 400065, China)

(Key Laboratory of Mobile Communication Technology, Chongqing University of Post and Telecommunications, Chongqing 400065, China)

Abstract: For the problems of low delay, low power requirement and access congestion caused by computational unloading of mass devices, a Joint Offloading Decision and Resource Allocation Algorithm (JODRAA) is proposed based on cloud-fog hybrid network architecture. Firstly, the algorithm considers the combination of cloud and fog computing, and establishes a resource optimization model to minimize system energy consumption and resource cost with maximum delay as constraint. Secondly, the original problem is transformed into a standard Quadratically Constrained Quadratic Program (QCQP) problem, and a low-complexity joint unloading decision-making and computational resource allocation algorithm is designed. Furthermore, considering the access congestion problem caused by massive computing of unloading devices, an estimation model of the overflow probability of unloading user access request queue is established, and an on-line measurement based time-frequency resource allocation algorithm for fog nodes is proposed. Finally, the iterative bandwidth and power allocation strategy is obtained by using fractional programming theory and Lagrange dual decomposition method. The simulation results show that the proposed algorithm can minimize the system energy consumption and resource cost on the premise of time delay.

Key words: Vehicular network; Fog computing; Computation offload; Resource allocation

收稿日期: 2019-04-30; 改回日期: 2019-12-13; 网络出版: 2020-07-01

*通信作者: 肖娇 Ir_xiao@163.com

基金项目: 国家自然科学基金(61571073), 重庆市教委科学技术研究项目(KJZD-M201800601)

Foundation Items: The National Natural Science Foundation of China (61571073), The Science and Technology Research Program of Chongqing Municipal Education Commission (KJZD-M201800601)

1 引言

雾计算通过将计算密集型任务卸载到云服务器，帮助移动设备克服资源限制。雾计算车联网是将雾计算范式扩展到传统蜂窝车联网网络中，可以实现更好的通信效率，降低无线网络运维成本^[1-3]。

文献[4,5]研究了多用户及多个雾节点场景中的任务卸载。文献[6]在多用户移动边缘网络场景下，提出一种基于模式枚举的优化方法对计算模式选择和系统传输时间分配进行联合优化。文献[7]考虑了云、雾计算服务器，研究了面向能耗优化的云雾混合计算卸载问题。文献[8,9]解决接入控制和服务用户的资源动态调度问题。文献[10]提出了一种雾计算辅助的拥塞避免方案。文献[11]研究了异构云无线接入网络环境下的联合拥塞控制和资源的随机优化问题。文献[12]提出了一种集成云雾计算的方法，以最小化系统总能耗及时延为目标，对卸载决策、计算资源和无线资源分配进行联合优化。

综上所述，前期大量文献分开研究卸载决策或资源分配，较少考虑联合优化卸载决策及所涉及的相关资源分配。并且，大量研究仅关注雾计算卸载的低时延、低能耗优势，未结合云服务器的高计算能力进行计算卸载协作以满足不同用户的QoS需求。此外，大量工作集中考虑高可靠、低时延通信需求，忽略了海量设备计算卸载引起的网络拥塞问题。针对以上不足，本文贡献如下：

首先，在云、雾计算混合的车联网架构下，针对车联网业务的低时延、低功耗需求及海量设备计算卸载的雾节点网络拥塞问题，建立联合计算卸载、计算资源和无线资源分配的资源优化模型，在满足时延需求的同时最小化系统能耗和资源成本。

其次，将原问题转化为标准QCQP问题，通过半定松弛和二分搜索方法设计一种低复杂度的联合卸载决策和计算资源分配算法。进一步针对海量设备计算卸载的雾节点接网络拥塞问题，利用大偏差原理建立推导出卸载用户请求队列的上溢概率，提出一种基于在线测量的雾节点时频资源配置算法。

最后，提出一种基于分式规划的无线资源分配算法，将非凸分式近似为凸减式优化问题，并利用拉格朗日对偶分解方法得到雾节点关联用户的带宽和功率分配方案。

2 系统模型与问题建模

2.1 无线通信模型

基于云雾混合计算的车联网网络架构如图1所示，大量雾节点 f 部署在用户终端 m 附近，不同用户根据

的计算卸载服务，移动终端通过无线接入链路或回程链路将计算任务的数据传输到雾节点或云服务器。

假设每个雾节点设备可以服务多个用户 $m \in M$ ， $m \in A_f$ 表示接入到雾节点设备 $f \in F$ 的用户集，因此雾节点为关联的用户 m 分配资源块和功率后可达到的传输速率为

$$r_m = \alpha_m B_{SC} \lg \left(1 + \frac{p_m^{\text{com}} h_m}{N_0 \alpha_m B_{SC}} \right) \quad (1)$$

其中 B_{SC} 表示单位资源块占用的带宽， B_f^{max} 为雾节点 f 拥有的最大带宽， p_m^{com} 和 h_m 分别表示雾节点 f 到其关联的用户 m 的传输功率和信道增益， N_0 表示噪声功率谱密度。

采用OFDMA技术， α_m 表示分配给用户 m 的资源块数目，带宽资源分配策略为

$$\alpha_m \in \{1, 2, \dots, B_f^{\text{max}}/B_{SC}\} \quad (2)$$

雾节点 f 的服务速率为 $r_f = \sum_{m \in M} r_m$ ，通过功率和频谱分配优化服务速率实现接入控制。假设 $A_n^f \in A = \{0, 1, \dots, s_A^f\}$ ， $D_n^f \in D = \{0, 1, \dots, s_D^f\}$ 分别表示在调度周期 n 内的业务请求到达和离开数量， Q_n^f 表示在第 n 个时隙开始时刻的队列长度，队列动态可以表示为 $Q_{n+1}^f = \max\{Q_n^f - D_n^f, 0\} + A_n^f$ 。为满足车辆接入雾节点的QoS需求，假设可容忍的接入时延为 d_f ，对应的队列阈值的上界为 $B_H^f = d_f \times r_f$ 。因此用户接入的QoS需求可以转化为队列上溢概率问题，其中 ε_f 为最大违反概率

$$P_{\text{of}}^f = P(Q_n^f > B_H^f) < \varepsilon_f, \quad f = 1, 2, \dots, F \quad (3)$$

2.2 计算卸载模型

由于本地处理能力有限，决策器需要根据卸载请求及无线信道增益选择在雾层或云层处理应用程序，用户 m 的卸载决策表示为

$$x_m^{\text{fog}} + y_m^{\text{cloud}} = 1, \quad \forall m \in M \quad (4)$$

其中 $x_m^{\text{fog}} = 1, y_m^{\text{cloud}} = 1$ ，表示应用程序分别卸载在

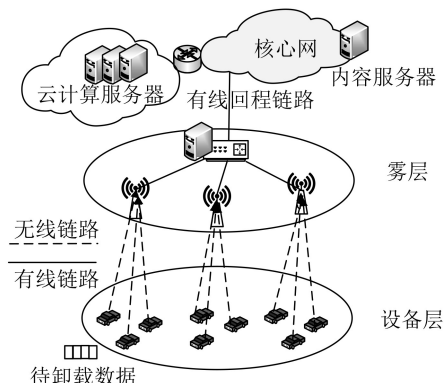


图1 云雾混合车联网网络架构

雾层、云层处理, 假设每个用户 m 携带 D_m bit数据, 相应的处理 D_m bit数据需要 C_m 个CPU周期

$$C_m = D_m \cdot u_m \quad (5)$$

其中 u_m 为处理密度, 假设 f_m^{fog} 和 p_m^{id} 分别为雾节点的计算能力和单位时间功率消耗, 用户需要通过无线链路传输待处理的数据, 传输速率为 r_m , 于是雾节点处理应用程序需要的时延和能耗为^[13,14]

$$T_m^{\text{fog}} = D_m/r_m + C_m/f_m^{\text{fog}} \quad (6)$$

$$E_m^{\text{fog}} = p_m^{\text{com}}(D_m/r_m) + p_m^{\text{id}}(C_m/f_m^{\text{fog}}) \quad (7)$$

由于雾层计算资源有限, 需要借助云层处理数据。假设用户 m 在有线链路的传输速率为 R_m^{fc} , 云层分配给用户 m 的计算能力为 f_m^{c} , 于是云服务器处理数据需要的总时延和能耗为

$$T_m^{\text{cloud}} = D_m/r_m + D_m/R_m^{\text{fc}} + C_m/f_m^{\text{c}} \quad (8)$$

$$E_m^{\text{cloud}} = p_m^{\text{com}}(D_m/r_m) + p_m^{\text{id}}(D_m/R_m^{\text{fc}} + C_m/f_m^{\text{c}}) \quad (9)$$

根据式(4), 处理用户 m 的应用程序需要时延和能耗可以表示为

$$T_m = T_m^{\text{fog}} x_m^{\text{fog}} + T_m^{\text{cloud}} y_m^{\text{cloud}} \quad (10)$$

$$E_m = E_m^{\text{fog}} x_m^{\text{fog}} + E_m^{\text{cloud}} y_m^{\text{cloud}} \quad (11)$$

ρ_m 和 ω_m 分别表示单位物理资源块和计算资源成本, 本文联合考虑资源成本和能耗, 定义系统效用函数为

$$U_m = (\rho_m \alpha_m B_{\text{SC}} + \omega_m f_m^{\text{fog}}) \cdot E_m \quad (12)$$

2.3 问题建模

本文提出一种云雾混合计算系统的联合卸载决策、计算资源和无线资源分配的优化模型, 在满足最大时延约束同时最小化系统效用函数, 建立模型为

$$\left. \begin{array}{l} \min_{\alpha, p, f^{\text{fog}}, \nu} \sum_{m \in M} U_m \\ \text{C1: } x_m^{\text{fog}} + y_m^{\text{cloud}} = 1, \forall m \in M \\ \text{C2: } x_m^{\text{fog}}, y_m^{\text{cloud}} \in \{0, 1\}, \forall m \in M \\ \text{C3: } P(Q_n^f > B_H^f) < \varepsilon_f, \forall f \in F \\ \text{C4: } \sum_{m \in M} \alpha_m \leq B_f^{\text{max}} \\ \text{C5: } 0 \leq p_m^{\text{com}} \leq p^{\text{max}}, \forall m \in M \\ \text{C6: } \sum_{m \in M} f_m^{\text{fog}} \leq F^{\text{fog}} \\ \text{C7: } T_m \leq d_m^{\text{max}}, \forall m \in M \end{array} \right\} \quad (13)$$

其中 F^{fog} 表示雾节点的总计算资源; 约束C1和C2表示用户的计算卸载决策; C3表示接入控制的上溢概率不能超过规定阈值; C4为用户的带宽资源分配约束; C5为用户的功率控制约束; C6表示分配

的计算资源和不能超过雾节点总的计算能力; C7表示每个应用的处理时间不超过最大可容忍时延。

3 联合计算卸载决策、接入控制和资源分配方案

3.1 联合计算卸载决策和计算资源分配方案

首先进行卸载决策及计算资源调度, 然后雾节点根据估计的队列溢出概率对资源块数量配置进行动态调整, 在最小化资源成本的同时实现接入控制, 最后在满足时延和资源总量约束下, 进行无线资源优化^[15]。

$$\left. \begin{array}{l} \min_{\alpha, p, f^{\text{fog}}, \nu} \sum_{m \in M} U_m \\ \text{s.t. } \text{C1, C2, C6} \end{array} \right\} \quad (14)$$

令 $\max_{m \in M} p_m^{\text{id}} \frac{C_m}{f_m^{\text{fog}}} x_m^{\text{fog}} = D_m^{\text{fog}}$, $\max_{m \in M} p_m^{\text{com}} \frac{D_m}{r_m} (y_m^{\text{fog}} + z_m^{\text{cloud}}) = D_m^{\text{com}}$, 为了减小计算复杂度, 将问题转化为二次约束二次规划(QCQP), 假设 $U_{\text{tot}} = \sum_{m \in M} U_m$, 定义向量

$$\mathbf{w} = [x_1^{\text{fog}}, y_1^{\text{cloud}}, \dots, x_M^{\text{fog}}, y_M^{\text{cloud}}, f_1^{\text{fog}}, \dots, f_M^{\text{fog}}, D_1^{\text{fog}}, \dots, D_M^{\text{fog}}, r_1, \dots, r_M, D_1^{\text{com}}, \dots, D_M^{\text{com}}, U_{\text{tot}}]^T \quad (15)$$

优化问题可以进一步转化为标准的非凸QCQP问题, 可分半定松弛(SDR)可以有效简化QCQP问题并得到近似的优化解。定义 $\mathbf{q} = [\mathbf{w}_{(6M+1) \times 1} \mathbf{1}_{1 \times 1}]^T$, $\mathbf{Q} = [\mathbf{q}\mathbf{q}^T]_{(6M+2) \times (6M+2)}$, \mathbf{Q} 为一个秩为1的半正定对称矩阵, 将式(14)转化为式(16)的等价形式

$$\left. \begin{array}{l} \min_{\mathbf{Q}} \text{Tr}(\mathbf{A}_0 \mathbf{Q}) \\ \text{s.t. } \text{C1}'' : \text{Tr}(\mathbf{A}_p \mathbf{Q}) = 0, p = 1, 2, \dots, 3M \\ \text{C2}'' : \text{Tr}(\mathbf{A}_m^L \mathbf{Q}) = 1, \forall m \in M \\ \text{C6}'' : \text{Tr}(\mathbf{A}^{\text{fog}} \mathbf{Q}) \leq F^{\text{fog}} \\ \text{C8}'' : \text{Tr}(\mathbf{A}_m^{\text{fog}} \mathbf{Q}) \leq 0, \forall m \in M \\ \text{C9}'' : \text{Tr}(\mathbf{A}_m^{\text{com}} \mathbf{Q}) \leq 0, \forall m \in M \\ \text{C10}'' : \mathbf{Q}(6M+2, 6M+2) = 1 \\ \text{C11}'' : \mathbf{Q} \geq 0 \\ \text{C12}'' : \text{rank}(\mathbf{Q}) = 1 \end{array} \right\} \quad (16)$$

在式(16)中, 只有约束C12''为非凸约束, 因此原优化问题可以松弛为式(17)正半定规划问题

$$\left. \begin{array}{l} \min_{\mathbf{Q}} \text{Tr}(\mathbf{A}_0 \mathbf{Q}) \\ \text{s.t. } \text{C1}'' \sim \text{C2}'', \text{C6}'', \text{C8}'' \sim \text{C11}'' \end{array} \right\} \quad (17)$$

联合卸载决策和计算资源分配算法: 根据 \mathbf{Q} 的定义, 通过提取优化解 \mathbf{Q}^* 左上方的 $2M \times 2M$ 子矩阵 \mathbf{Q}'^* , 可以得到卸载决策方案 \mathbf{v} 。 $\mathbf{Pr} = [\text{pr}_1^f, \text{pr}_1^c, \dots, \text{pr}_m^f, \text{pr}_m^c]^T \triangleq \text{diag}(\mathbf{Q}'^*)$, 其中 \mathbf{Pr}

中的每个元素代表卸载决策中对应元素为1的概率，为了满足约束C2，定义 $\varpi_m^f = \text{pr}_m^f(1 - \text{pr}_m^c)$ ， $\varpi_m^c = (1 - \text{pr}_m^f)\text{pr}_m^c$ ，对应用户 m 的决策概率为 $\text{pr}_m^f = \varpi_m^f / (\varpi_m^f + \varpi_m^c)$ ， $\text{pr}_m^c = \varpi_m^c / (\varpi_m^f + \varpi_m^c)$ ，用户 m 任务卸载决策为

$$\kappa_m = \left\{ \begin{array}{l} (1, 0), \text{以概率}\text{pr}_m^f \text{进行雾计算} \\ (0, 1), \text{以概率}\text{pr}_m^c \text{进行云计算} \end{array} \right\} \quad (18)$$

根据式(18)中的概率随机设置向量值，得到用户 m 的卸载决策方案 x_m^{fog} ， y_m^{cloud} ，由于卸载决策是根据对应的概率随机获得的，通常多次迭代上述过程以获得准确的卸载决策。在经过 J 次卸载决策 \mathbf{v}^j 之后，在对应的卸载决策方案下，进行计算和无线资源分配，最终选择使目标函数最小化的卸载决策方案 \mathbf{v}^* 。

在得到计算卸载决策后，首先进行计算资源分配，已知 $p_m^{\text{id}} \left(\frac{D_m}{R_m^{\text{fc}}} + \frac{C_m}{f_m^c} \right)$ 和无线资源分配，则 $S_m = p_m^{\text{com}} \frac{D_m}{r_m}$ 也恒定，令C13： $p_m^{\text{id}} \frac{C_m}{f_m^{\text{fog}}} + S_m \leq \chi$ ，原问题可以转化为

$$\left. \begin{array}{l} \min_{\alpha, p, f^{\text{fog}}, \nu} \sum_{m \in M} E_m \\ \text{s.t. C6, C13} \end{array} \right\} \quad (19)$$

由于 $p_m^{\text{id}} \frac{C_m}{f_m^{\text{fog}}} \geq 0$ ，于是有 $\chi - S_m \geq 0$ ，因此 $0 \leq \frac{C_m p_m^{\text{id}}}{\chi - S_m} \leq f_m^{\text{fog}}, \forall m \in M$ ，可以得到 $\sum_{m \in M} \frac{C_m p_m^{\text{id}}}{\chi - S_m} \leq \sum_{m \in M} f_m^{\text{fog}} \leq F^{\text{fog}}$ 。接着执行雾节点的计算资源调度，为能量成本较大的用户分配更多的资源，因此该用户的能量成本降低而其他用户的能量成本增加，通过迭代执行该过程知道所有计算资源分配完毕，并且雾节点将分配相同质量的计算资源给所有用户，于是有

$$\sum_{m \in M} \frac{C_m p_m^{\text{id}}}{\chi - S_m} = \sum_{m \in M} f_m^{\text{fog}} = F^{\text{fog}} \quad (20)$$

因此式(20)可以转化为

$$\left. \begin{array}{l} \min_{\chi} \chi \\ \text{s.t. } \sum_{m \in M} \frac{C_m p_m^{\text{id}}}{\chi - S_m} = \sum_{m \in M} f_m^{\text{fog}} = F^{\text{fog}} \end{array} \right\} \quad (21)$$

由于上述约束的左侧随 χ 单调递减，可以采用二分搜索法求解，因此卸载决策和计算资源调度算法如表1。

3.2 基于在线测量的接入管理

最大可容忍时延内，雾节点可能无法服务全部队列中的接入请求。因此，雾节点需要根据卸载决

策在线动态地配置更多的时频资源块以提高服务速率避免海量接入引起的拥塞。

3.2.1 上溢概率评估模型

在给定的周期 n 上，对应的队列增量为 $I_n^f = A_n^f - D_n^f$ ，其中 I_n^f 的取值空间为 $I_n^f \in \{-s_D^f, \dots, 0, \dots, s_A^f\}$ 。在第 n 到第 $n+N$ 个周期内，雾节点 f 的队列增量可表示为 $T_{n+N}^f = \sum_{i=1}^N I_{n+i}^f$ 。在第 $n+N$ 个周期开始时，雾节点 f 的瞬时队列长度为 $Q_{n+N}^f = Q_n^f + T_{n+N}^f$ 。因此可得雾节点 f 在第 $n+N$ 个周期上的队列上溢概率为

$$\begin{aligned} P_{n+N}^f &= P(Q_{n+N}^f > B_H^f) = P(Q_n^f + T_{n+N}^f > B_H^f) \\ &= P\left(\frac{T_{n+N}^f}{N} > \frac{B_H^f - Q_n^f}{N}\right) \\ &= P\left(\frac{\sum_{i=1}^N I_{n+i}^f}{N} > a_0^f\right) \end{aligned} \quad (22)$$

其中 $a_0^f = (B_H^f - Q_n^f)/N$ 表示雾节点 f 在未来的 N

表 1 联合卸载决策和基于二分法的计算资源调度算法

1. 初始化试验次数 J ，用户数 M ，总带宽 B_j^{max} 及资源块带宽 B_{SC} 及总计算资源 F^{fog} ，初始化用户参数 $D_m, u_m, f_m^{\text{loc}}, p_m^{\text{max}}, p_m^{\text{id}}, p_m^{\text{loc}}, R_m^{\text{fc}}, f_m^c, d_m^{\text{max}}$ ，初始化式(17)中的所有矩阵
2. 利用凸优化工具求解式(17)得到优化解 \mathbf{Q}^*
3. 从优化解 \mathbf{Q}^* 中提取左上角 $2M \times 2M$ 的子矩阵 \mathbf{Q}'^* ， \mathbf{Q}'^* 中的对角线上的元素值为 $\text{Pr} = [\text{pr}_1^f, \text{pr}_1^c, \dots, \text{pr}_M^f, \text{pr}_M^c]$
4. for $j = 1; j \leq J; j++$ do
5. 根据式(18)从 Pr 中提取卸载决策 \mathbf{v}^j
6. 执行计算资源调度：初始化参数 $\chi^{\text{min}} = \max\{S_m\}$ ， $\chi^{\text{max}} = \sum_m \left(\frac{C_m p_m^{\text{id}} M}{F^{\text{fog}}} + S_m \right)$ ，于是有 $\chi^{\text{min}} \leq \chi^{\text{opt}} \leq \chi^{\text{max}}$ ，最大可容忍误差 $\varepsilon > 0$ ， $\chi^j = (\chi^{\text{min}} + \chi^{\text{max}})/2$
7. while $|\chi^{\text{max}} - \chi^{\text{min}}| \geq \varepsilon$ do
8. if $\sum_{m \in M} \frac{C_m p_m^{\text{id}}}{\chi^j - S_m} > F^{\text{fog}}$ then
9. $\chi^{\text{min}} = \chi^j$
10. else
11. $\chi^{\text{max}} = \chi^j$
12. end if
13. end while
14. if $|\chi^{\text{max}} - \chi^{\text{min}}| \leq \varepsilon$ then
15. $\chi^{\text{opt}} = \chi^j$
16. end if
17. 将得到的 χ^{opt} 代入式(21)得到计算资源调度策略 \mathbf{f}^{fog}
18. end for

个周期内可接受的平均队列增长率, 令 $m_o^f = E \left[\sum_{i=1}^N I_{n+i}^f / N \right]$ 表示雾节点 f 在未来 N 个周期内期望的平均队列增长率, $E[\cdot]$ 是期望因子。在 $m_o^f < a_o^f$ 的情况下, 在 N 个周期之后队列溢出的可能性非常大, 应该为雾节点配置更多的资源块。当 $m_o^f > a_o^f$ 时, 尽管式(22)中的概率 P_{n+N}^f 较小, 但仍有可能超过阈值 ε_f 。当 $P_{n+N}^f > \varepsilon_f$ 时, 雾节点将控制用户的服务速率^[16]。

假设 A_n^f 是独立同分布过程, 则 I_n^f 同样也是独立同分布的随机变量, 且满足一个有限瞬时生成的函数 $M(\theta) = E \left[e^{\theta I_n^f} \right]$ 。若 $E \left[e^{\theta I_n^f} \right] > a_o^f$, 队列 I_n^f 遵循大偏差原则, 因此, 当 $a_o^f > m_o^f$, 根据Cramer's 定理可得

$$\lim_{N \rightarrow \infty} \frac{1}{N} \lg P \left(\left(\sum_{i=1}^N I_{n+i}^f \right) / N > a_o^f \right) = -L(a_o^f) \quad (23)$$

式中 $L(a_o^f)$ 为速率函数, 具体表示为 $L(a_o^f) = \sup_{\theta > 0} \{ a_o^f \theta - \lg M(\theta) \}$, $\lg M(\theta) = \lg \left\{ \sum_{i=-s_D}^{s_A} \pi_i \exp[i\theta] \right\}$ 。

根据式(23), 当 N 较大时, 上溢概率可近似为

$$P_{n+N}^f \approx \exp(-NL(a_o^f)) \quad (24)$$

由于 a_o^f , m_o^f 和 π_i^f 未知, 因此接下来采用一种基于滑动窗口的方法来在线估计上述参数。

3.2.2 上溢概率在线估计

可以通过滑动窗口获取历史观测结果, 假设滑动周期为 N_ω , 在雾节点 f 观测到的队列长度增量为 $\{I_1^f, I_2^f, \dots\}$, 滑动窗口在改序列上移动并覆盖最新的 N_ω 个条目, 对于当前周期 n , 雾节点 f 的观测向量可表示为 $\mathbf{W}_n^f = [I_n^f, I_{n-1}^f, \dots, I_{n-N_\omega+1}^f]$, 可以通过历史观测的平均值来估计 m_o^f , 即 $\hat{m}_o^f = \left(\sum_{i=n-N_\omega+1}^n I_i^f \right) / N_\omega$, 将 N_i^f 定义为滑动窗口中出现 $I_n^f = i$ 的事件, $I_n^f = i$ 的对应频率可以估计 $\hat{q}_i^f(n) = N_i^f / N_\omega$ 。若直接采用 $\hat{q}_i^f(n)$ 作为 π_i^f 的估计值可能造成 π_i^f 在不同周期上出现较大的波动, 可利用指数平滑去柔和队列增量概率的估计值, 表示为

$$\hat{\pi}_i^f(n) = \rho \hat{\pi}_i^f(n-1) + (1-\rho) \hat{q}_i^f(n) \quad (25)$$

表2为雾节点关联用户基于在线测量的接入控制算法。

3.3 迭代的无线资源分配算法

在得到计算资源调度方案 \mathbf{f}^{fog} 后, 剩余问题为优化带宽和功率资源, 原问题转化为

$$\min_{\alpha, \mathbf{p}} \left. \frac{D_m p_m^{\text{com}}}{\alpha_m B_{\text{SC}} \lg \left(1 + \frac{p_m^{\text{com}} h_m}{N_0 \alpha_m B_{\text{SC}}} \right)} \right\} \quad (26)$$

s.t. C4, C5, C7

式(26)可以分为非线性分式规划问题, 对应的优化解为 $\{\mathbf{p}^*, \alpha^*\}$ 和 V^* , 其中 $V^* =$

$$\min_{\mathbf{p}, \alpha} \frac{D_m p_m^{\text{com}*}}{\alpha_m^* B_{\text{SC}} \lg \left(1 + \frac{p_m^{\text{com}*} h_m}{N_0 \alpha_m^* B_{\text{SC}}} \right)}, \quad \text{当且仅当}$$

式(27)成立, 可以得到优化值 V^*

$$\min_{\mathbf{p}^*, \alpha^*} \left[D_m p_m^{\text{com}*} - V^* \alpha_m^* B_{\text{SC}} \lg \left(1 + \frac{p_m^{\text{com}*} h_m}{N_0 \alpha_m^* B_{\text{SC}}} \right) \right] = 0 \quad (27)$$

式(26)可等价于式(27), 由于 V^* 通常是未知的, 因此需要通过多次迭代更新参数 V , 优化问题为

$$\min_{\mathbf{p}, \alpha} \left. D_m p_m^{\text{com}} - V \alpha_m B_{\text{SC}} \lg \left(1 + \frac{p_m^{\text{com}} h_m}{N_0 \alpha_m B_{\text{SC}}} \right) \right\} \quad (28)$$

s.t. C4, C5, C7

将约束 C4 和 C5 代入 C7, 可以得到 C14: $\alpha_m B_{\text{SC}} \lg \left(1 + \frac{p_m^{\text{com}} h_m}{N_0 \alpha_m B_{\text{SC}}} \right) \geq \frac{D_m}{d_m^{\text{max}} - \tau_m}$, $\tau_m = p_m^{\text{id}} (C_m / f_m^{\text{fog}}) x_m^{\text{fog}} + p_m^{\text{id}} (D_m / R_m^{\text{fc}} + C_m / f_m^{\text{fc}}) y_m^{\text{cloud}}$, 令 C15: $D_m p_m^{\text{com}} - V \alpha_m \lg \left(1 + \frac{p_m^{\text{com}} h_m}{N_0 \alpha_m B_{\text{SC}}} \right) \leq \xi$, 因此式(28)松弛为

$$\min_{\mathbf{p}, \alpha, \xi} \left. \xi \right\} \quad (29)$$

s.t. C4, C5, C14, C15

表2 基于在线测量的接入控制算法

1. 初始化每个雾节点的资源块配置数量 z 和剩余资源块数量 B , 在周期 n 上观察每个雾节点 f 的接入请求队列状态 Q_n^f
2. for $f = 1; f < F; f++$ do
3. 计算 a_o^f , 估计 \hat{m}_o^f
4. while $Q_n^f \geq B_H^f$ or $B = \emptyset$ do
5. $z \leftarrow z + 1$, $C_f(n) \leftarrow z \bar{r}$, $B \leftarrow B - 1$
6. end while
7. 计算 a_o^f 及 \hat{m}_o^f
8. if $Q_n^f < B_H^f$ & $\hat{m}_o^f \geq a_o^f$ then
9. $z \leftarrow z + \Delta_1$, $C_f(n) \leftarrow z \bar{r}$, $B \leftarrow B - \Delta_1$
10. else if $Q_n^f < B_H^f$ & $\hat{m}_o^f \geq a_o^f$ then
11. 式(24)执行黄金分割搜索算法估计 \hat{P}_{n+N}^f
12. if $\hat{P}_{n+N}^f \geq \varepsilon_f$ then
13. $z \leftarrow z + \Delta_2$, $C_f(n) \leftarrow z \bar{r}$, $B \leftarrow B - \Delta_2$
14. end if
15. end if
16. end if
17. end for

式(29)为联合凸优化问题，因此可以通过拉格朗日对偶和次梯度方法求解，拉格朗日函数为

$$\begin{aligned}
D(\beta, \eta, \mu, \pi) &= \min_{\mathbf{p}, \alpha, \xi} L(\mathbf{p}, \alpha, \xi, \beta, \eta, \mu, \pi) \\
&= \xi + \beta \left(\sum_{m \in M} \alpha_m \leq B_f^{\max} \right) \\
&\quad + \sum_{m \in M} \eta_m (p_m^{\text{com}} - p^{\max}) \\
&\quad + \sum_{m \in M} \mu_m \left[\frac{D_m}{d_m^{\max} - \tau_m} \right. \\
&\quad \left. - \alpha_m B_{\text{SC}} \lg \left(1 + \frac{p_m^{\text{com}} h_m}{N_0 \alpha_m B_{\text{SC}}} \right) \right] \\
&\quad + \sum_{m \in M} \pi_m \left[D_m p_m^{\text{com}} - V \alpha_m B_{\text{SC}} \right. \\
&\quad \left. \cdot \lg \left(1 + \frac{p_m^{\text{com}} h_m}{N_0 \alpha_m B_{\text{SC}}} \right) - \xi \right] \quad (30)
\end{aligned}$$

根据KKT条件，分别对功率 p_m^{com} 和带宽 α_m 求导等于零，得到用户优化的功率解

$$\begin{aligned}
p_m^{\text{com}*} &= \frac{\partial D(\beta, \eta, \mu, \pi)}{\partial p_m^{\text{com}}} \\
&= \left\{ \alpha_m \left[\frac{B_{\text{SC}}(\mu_m + V\pi_m)}{\ln 2(\eta_m + \pi_m D_m)} - \frac{N_0 B_{\text{SC}}}{h_m} \right] \right\}^+, \\
&\quad \forall m \in M \quad (31)
\end{aligned}$$

在得到功率优化解后，通过对 α_m 求导并等于零，可以得到

$$\begin{aligned}
(\mu_m + V\pi_m) B_{\text{SC}} \lg \left(1 + \frac{p_m^{\text{com}} h_m}{\alpha_m N_0 B_{\text{SC}}} \right) \frac{\mu_m + V\pi_m}{\ln 2} \\
\cdot \frac{p_m^{\text{com}} h_m B_{\text{SC}}}{\alpha_m N_0 B_{\text{SC}} + p_m^{\text{com}} h_m} = \beta \quad (32)
\end{aligned}$$

假设 $\varphi_m = \varphi_m(\mu_m, \pi_m, \beta) = 1 / (1 + p_m^{\text{com}} h_m / \alpha_m N_0 B_{\text{SC}})$ ，式(32)可以转化为

$$\varphi_m - \ln 2 \cdot \lg(\varphi_m) = \frac{\beta \ln 2}{(\mu_m + V\pi_m) B_{\text{SC}}} + 1 \quad (33)$$

通过二分搜索法求得最优 φ_m^* ，因此对 α_m 求导得到的带宽分配优化解为

$$\alpha_m^* = \frac{\varphi_m^*(\mu_m, \pi_m, \beta) p_m^{\text{com}} h_m}{1 - \varphi_m^*(\mu_m, \pi_m, \beta) N_0 B_{\text{SC}}} \quad (34)$$

由于 $D_m p_m^{\text{com}} - V \alpha_m B_{\text{SC}} \lg \left(1 + \frac{p_m^{\text{com}} h_m}{\alpha_m N_0 B_{\text{SC}}} \right) \leq \xi \leq 0$ ，因此得到关于 ξ 的优化解为

$$\xi^* = \begin{cases} 0, \sum_{m \in M} \pi_m > 1 \\ \max_{m \in M} \left[D_m p_m^{\text{com}*} - V \alpha_m^* B_{\text{SC}} \right. \\ \left. \lg \left(1 + \frac{p_m^{\text{com}*} h_m}{N_0 \alpha_m^* B_{\text{SC}}} \right) \right], \sum_{m \in M} \pi_m < 1 \end{cases} \quad (35)$$

式(32)和式(35)求解依赖于拉格朗日乘子的变化，通过次梯度方法对拉格朗日乘子进行更新。表3为联合带宽和功率优化算法。

4 性能仿真与结果分析

本文根据表4给出的仿真参数利用MATLAB工具进行仿真，本节对所提算法进行验证，并将所提联合计算卸载和资源分配算法(Joint Offloading Decision and Resource Allocation Algorithm, JODRAA)分别与文献[5]仅进行计算卸载决策优化的Offloading-only算法、文献[12]的JOSRAA算法进行对比。

图2显示了节省能量随时延阈值的变化，可以看出所提的JODRAA算法在3种算法中性能最优，由于同时对资源分配和计算卸载进行优化，可以节省更多的能量。随着时延阈值的增大，也可以节省更多能量，这是因为在更宽松的时延阈值条件下，可以对更多用户进行卸载决策和资源优化，也可以节省更多能量。

图3描述了平均资源成本随雾节点数量的变化

表3 基于迭代的带宽和功率资源调度

1. 初始化迭代次数 $N_1=0$ 和 $N_2=0$ ，误差精度 δ_1 和 δ_2 ， $V^{N_1}=1$
2. while $N_1 < N_{1 \max}$ do
3. while $N_2 < N_{2 \max}$ do
4. 对给定的 V^{N_1} ，根据式(31)求得优化的传输功率解
5. 在区间 $[0, 1]$ 内执行二分搜索方法求解 $\varphi_m(N_2)$ ，并将 $\varphi_m(N_2)$ 代入式(34)求解带宽资源调度方案
6. 通过次梯度法分别更新拉格朗日乘子
7. if $\|\beta(N_2 + 1) - \beta(N_2)\|_2 < \delta_2$,
 $\|\eta(N_2 + 1) - \eta(N_2)\|_2 < \delta_2$,
 $\|\mu(N_2 + 1) - \mu(N_2)\|_2 < \delta_2$,
 $\|\pi(N_2 + 1) - \pi(N_2)\|_2 < \delta_2$ then
8. $\alpha_m^{N_1} = \alpha_m(N_2)$, $p_m^{\text{com}N_1} = p_m^{\text{com}}(N_2)$, break
9. else
10. $N_2 = N_2 + 1$
11. end if
12. end while
13. if $\left| D_m p_m^{\text{com}N_1} - V^{N_1} \alpha_m^{N_1} \lg \left(1 + \frac{p_m^{\text{com}N_1} h_m}{\alpha_m^{N_1} N_0 B_{\text{SC}}} \right) \right| < \delta_1$ then
14. $\{\mathbf{p}^*, \boldsymbol{\alpha}^*\} = \{\mathbf{p}^{\text{com}N_1}, \boldsymbol{\alpha}^{N_1}\}$
15. else
16. 令 $V^{N_1+1} = D_m p_m^{\text{com}N_1} / \alpha_m^{N_1} B_{\text{SC}} \lg \left(1 + \frac{p_m^{\text{com}N_1} h_m}{\alpha_m^{N_1} N_0 B_{\text{SC}}} \right)$
17. end if
18. end while
19. 输出无线资源调度优化解 $\mathbf{p}^*, \boldsymbol{\alpha}^*$

情况，从图中可以看出，随着雾节点数量的增加，平均资源成本呈近似线性增加，这是因为雾节点数

表4 仿真参数

| 参数 | 数值 |
|---------|-----------------------------|
| 系统带宽 | 10 MHz(50PRBs) |
| 路径损耗模型 | UrbanMicro(UMi) |
| 最大传输功率 | 23 dBm |
| 计算资源单价 | 0.10, 0.15, 0.20 unit/cycle |
| 计算密度 | 297.62 cycle/bit |
| 链路传输速率 | 1 Mb/s |
| 参数 | 数值 |
| 卸载业务到达 | 泊松分布 |
| 莱斯因子 | 6 dB |
| 滑动窗口大小 | 60 ms |
| 平滑指数 | 0.7 |
| 雾计算资源量 | 1 G cycle |
| 云层计算能力 | 2 G cycle/s |
| 参数 | 数值 |
| 比特到达速率 | 0.4 Mbit/ms |
| 噪声功率 | -174 dBm/Hz |
| PRB单价 | 1, 1.5, 2 unit/PRB |
| 仿真时间 | 6000 ms |
| 队列上溢概率 | 0.2 |
| 单位t功率消耗 | 0.01 W |

量增加，可以卸载更多的用户数据，对应也消耗更多的资源。本文提出的联合计算卸载和资源分配优化算法JODRAA和JOSRAA算法都明显优于仅进行卸载决策的优化算法，这是因为资源成本优化不仅跟卸载决策有关，还很大程度由资源分配算法决定。

图4描述了系统总能量消耗随雾节点用户数的变化情况，将雾节点接入用户数设置在60以内，在队列上溢概率不溢出情况下，观察能量消耗变化趋势。可以看出，总能量消耗随用户数的增加近似线性增加，JODRAA算法通过同时对计算卸载和资源分配进行优化，在3种算法中性能最佳，总能量消耗最少。图5描述了平均时延随雾节点数量的变化情况，可以看出，JODRAA算法时延性能表现最佳，而仅进行计算卸载优化的算法由于未进一步对资源分配进行优化以减少接入时延，因此时延性能最差。并且3种算法平均时延均随雾节点数量的增加而逐渐减小，这是因为雾节点数量增加，相应的计算资源和无线资源等也增加，可以使更多用户更快进行数据卸载，因此平均时延降低。图6描述了违反概率随卸载用户数的变化情况，从图中可以看出，当用户数小于40时，违反概率为0，这是因为用户数较少，且资源量充足。而当用户数逐渐增大时，违反概率呈近似线性逐渐增大，后逐渐收敛趋于稳定，且最大违反概率不超过30%。

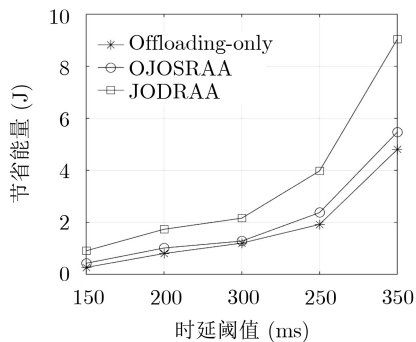


图2 节省能量与时延阈值的关系

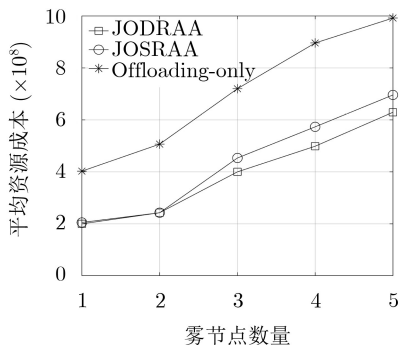


图3 平均资源成本与雾节点数量的关系

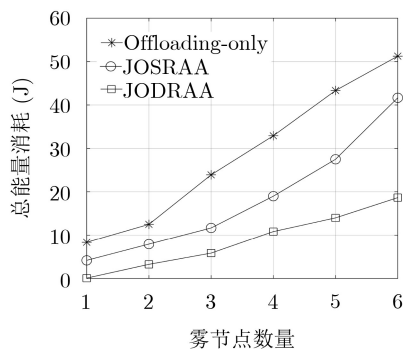


图4 总能量消耗与雾节点用户数的关系

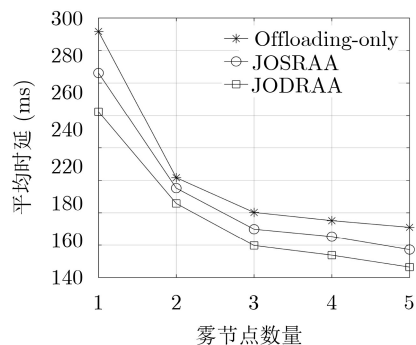


图5 平均时延与雾节点数量的关系

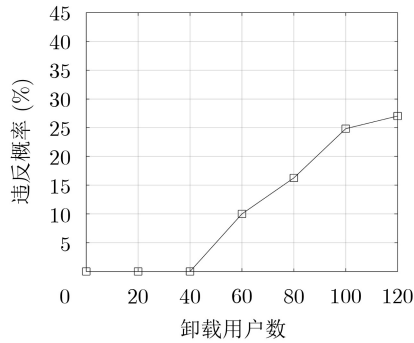


图6 违反概率与卸载用户数的关系

5 结束语

本文提出了在云雾混合网络架构下的联合计算卸载和资源分配算法JODRAA，在保证最大可容忍时延前提下，通过联合对卸载决策、计算资源、无线资源分配进行联合优化，解决了云雾混合系统中的计算卸载问题。接着建立一种在线测量模型，对雾节点时频资源进行配置，以减少计算卸载引起的网络拥塞。最后借助分式规划理论将非凸分式近似为凸减式优化问题，通过拉格朗日对偶分解方法得到雾节点关联用户的带宽和功率分配策略。仿真结果表明，本文所提算法可以在满足时延需求的前提下，最小化系统效用函数。

参考文献

- [1] MEBREK A, MERGHEM-BOULAHIA L, and ESSEGHIR M. Efficient green solution for a balanced energy consumption and delay in the IoT-Fog-Cloud computing[C]. The 16th IEEE International Symposium on Network Computing and Applications, Cambridge, USA, 2017: 1–4. doi: [10.1109/NCA.2017.8171359](https://doi.org/10.1109/NCA.2017.8171359).
- [2] BACCARELLI E, NARANJO P G V, SCARPINITI M, et al. Fog of everything: Energy-efficient networked computing architectures, research challenges, and a case study[J]. *IEEE Access*, 2017, 5: 9882–9910. doi: [10.1109/ACCESS.2017.2702013](https://doi.org/10.1109/ACCESS.2017.2702013).
- [3] LIU Kaiyang, PENG Jun, ZHANG Xiaoyong, et al. A combinatorial optimization for energy-efficient mobile cloud offloading over cellular networks[C]. 2016 IEEE Global Communications Conference, Washington, USA, 2016: 1–6. doi: [10.1109/GLOCOM.2016.7841488](https://doi.org/10.1109/GLOCOM.2016.7841488).
- [4] YANG Lei, CAO Jiannong, TANG Shaojie, et al. A framework for partitioning and execution of data stream applications in mobile cloud computing[C]. The 5th IEEE International Conference on Cloud Computing, Honolulu, USA, 2012: 794–802. doi: [10.1109/CLOUD.2012.97](https://doi.org/10.1109/CLOUD.2012.97).
- [5] LIU Mengyu and LIU Yuan. Price-based distributed offloading for mobile-edge computing with computation capacity constraints[J]. *IEEE Wireless Communications Letters*, 2018, 7(3): 420–423. doi: [10.1109/LWC.2017.2780128](https://doi.org/10.1109/LWC.2017.2780128).
- [6] CAO Xiaowen, WANG Feng, XU Jie, et al. Joint computation and communication cooperation for energy-efficient mobile edge computing[J]. *IEEE Internet of Things Journal*, 2019, 6(3): 4188–4200. doi: [10.1109/JIOT.2018.2875246](https://doi.org/10.1109/JIOT.2018.2875246).
- [7] MENG Xianling, WANG Wei, and ZHANG Zhaoyang. Delay-constrained hybrid computation offloading with cloud and fog computing[J]. *IEEE Access*, 2017, 5: 21355–21367. doi: [10.1109/ACCESS.2017.2748140](https://doi.org/10.1109/ACCESS.2017.2748140).
- [8] GU H Y, YANG C Y, and FONG B. Low-complexity centralized joint power and admission control in cognitive radio networks[J]. *IEEE Communications Letters*, 2009, 13(6): 420–422. doi: [10.1109/LCOMM.2009.082173](https://doi.org/10.1109/LCOMM.2009.082173).
- [9] JIANG Menglan, CONDOLUCI M, and MAHMOODI T. Network slicing management & prioritization in 5G mobile systems[C]. The 22th European Wireless Conference, Oulu, Finland, 2016: 1–6.
- [10] YAQOOB S, ULLAH A, AKBAR M, et al. Fog-assisted congestion avoidance scheme for internet of vehicles[C]. The 14th International Wireless Communications & Mobile Computing Conference, Limassol, Cyprus, 2018: 618–622. doi: [10.1109/IWCMC.2018.8450402](https://doi.org/10.1109/IWCMC.2018.8450402).
- [11] LI Jian, PENG Mugen, YU Yuling, et al. Energy-efficient joint congestion control and resource optimization in heterogeneous cloud radio access networks[J]. *IEEE Transactions on Vehicular Technology*, 2016, 65(12): 9873–9887. doi: [10.1109/TVT.2016.2531184](https://doi.org/10.1109/TVT.2016.2531184).
- [12] LIU Yiming, YU F R, LI Xi, et al. Distributed resource allocation and computation offloading in fog and cloud networks with non-orthogonal multiple access[J]. *IEEE Transactions on Vehicular Technology*, 2018, 67(12): 12137–12151. doi: [10.1109/TVT.2018.2872912](https://doi.org/10.1109/TVT.2018.2872912).
- [13] LI Qiuping, ZHAO Junhui, GONG Yi, et al. Energy-efficient computation offloading and resource allocation in fog computing for internet of everything[J]. *China Communications*, 2019, 16(3): 32–41.
- [14] SHAHZADI R, NIAZ A, ALI M, et al. Three tier fog networks: Enabling IoT/5G for latency sensitive applications[J]. *China Communications*, 2019, 16(3): 1–11.
- [15] SOOKHAK M, YU F R, HE Ying, et al. Fog vehicular computing: Augmentation of fog computing using vehicular cloud computing[J]. *IEEE Vehicular Technology Magazine*, 2017, 12(3): 55–64. doi: [10.1109/MVT.2017.2667499](https://doi.org/10.1109/MVT.2017.2667499).
- [16] LI Di, KAR S, and CUI Shuguang. Distributed quickest detection in sensor networks via two-layer large deviation analysis[J]. *IEEE Internet of Things Journal*, 2018, 5(2): 930–942. doi: [10.1109/JIOT.2018.2810825](https://doi.org/10.1109/JIOT.2018.2810825).

唐 伦：男，1973年生，教授，博士生导师，主要研究方向为新一代无线通信网络、异构蜂窝网络等。

肖 娇：女，1995年生，硕士生，研究方向为蜂窝车联网下的资源调度算法。

魏延南：男，1995年生，硕士生，研究方向为5G网络切片、虚拟资源分配、随机优化理论。

赵国繁：女，1993年生，硕士生，研究方向为5G网络切片中的资源分配，可靠性。

陈前斌：男，1967年生，教授，博士生导师，主要研究方向为个人通信、多媒体信息处理与传输、下一代移动通信网络、异构蜂窝网络等。

责任编辑：陈 倩