



专题：面向6G的通感算一体化

基于通信-感知-计算一体化的高能效数据处理设计

周梓钦¹, 周尚语¹, 韩子栋¹, 曹晓雯², 李晓阳¹

(1. 南方科技大学工学院, 广东 深圳 518055;

2. 深圳大学电子与信息工程学院, 广东 深圳 518060)

摘要: 在物联网中, 传统的数据处理设计将感知、通信和计算过程分开考虑, 导致了无线电、能源和算力资源的严重浪费。为克服这一缺陷, 提出了一种通信-感知-计算一体化设计。该设计旨在根据移动设备和服务器的算力配置, 对数据感知速率、卸载量和卸载速率进行联合优化, 从而实现高能效的数据处理。研究表明, 数据卸载率由服务器的算力配置决定, 同时基于拉线算法的感知和卸载速率控制可达到能效最优。上述设计的有效性通过仿真进行了验证。

关键词: 通信-感知-计算一体化; 物联网; 数据卸载; 能效; 拉线算法

中图分类号: TN919; TP393

文献标志码: A

doi: 10.11959/j.issn.1000-0801.2026042

Integrated communication-sensing-computation based energy efficient data processing design

Zhou Ziqin¹, Zhou Shangyu¹, Han Zidong¹, Cao Xiaowen², Li Xiaoyang¹

1. College of Engineering, Southern University of Science and Technology, Shenzhen 518055, China

2. College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518060, China

Abstract: Conventional data processing in IoT often treats sensing, communication, and computation as separate tasks, leading to significant inefficiencies in radio, energy, and computational resource usage. To address this limitation, an integrated communication-sensing-computation design was proposed. This approach jointly optimized the data offloading ratio, sensing rate, and transmission rate based on the processing capabilities of both mobile devices and servers, with the goal of maximizing energy efficiency. The analysis revealed that the optimal offloading ratio depends solely on the server's processor profile. A string-pulling algorithm was further designed to determine the opti-

收稿日期: 2025-10-03; 修回日期: 2025-12-09

通信作者: 李晓阳, lixy@sustech.edu.cn

基金项目: 中国科协青年托举工程项目 (No.YESS20240364); 广东省未来智联网重点实验室项目 (No.2022B1212010001); 深圳市自然科学基金资助项目 (No.JCYJ20220530113017039, No.JCYJ20241202124934046, No.RCBS20231211090520032)

Foundation Items: The Young Elite Scientists Sponsorship Program by CAST (No.YESS20240364), The Guangdong Provincial Key Laboratory of Future Networks of Intelligence (No.2022B1212010001), The Shenzhen Science and Technology Program (No.JCYJ20220530113017039, No.JCYJ20241202124934046, No.RCBS20231211090520032)



mal sensing and offloading rates. Simulation results validate the superiority of the proposed design.

Key words: integrated communication-sensing-computation, Internet of things, data offloading, energy efficiency, string-pulling algorithm

0 引言

物联网应用需要终端设备感知海量数据，而终端设备有限的计算能力难以处理所有感知到的数据^[1]。因此，部分感知数据需要卸载到边缘服务器进行进一步处理，这被称为移动边缘计算^[2]。移动边缘计算的主要功能是将移动计算、网络管理和存储能力扩展到网络边缘，从而为资源有限的设备上需要密集计算和严苛时延的应用提供支持^[3]。移动边缘计算关注整合无线和算力资源，对计算卸载和网络架构进行各种创新设计^[4]。然而，大多数现有的移动边缘计算方案仅关注数据卸载和计算过程，并假设数据已被感知。实际上，数据感知是一个随终端设备工作条件变化的时变过程，也需要进行设计。

传统的数据处理设计将感知、通信和计算过程分开考虑，导致了无线电、能源和计算资源的严重浪费。近年来，先进的通信感知一体化技术显著节省了无线资源^[5]。作为一项前沿技术，通信感知一体化技术通过复用无线信号，在统一的系统平台和共享的信号频谱上实现了同时感知与通信^[6]。然而，通信感知一体化的研究未考虑数据的计算过程，导致数据处理效率受限。

作为通信感知一体化的演进，通信-感知-计算一体化技术将数据计算过程进一步纳入系统设

计^[7]。沿着这一思路，学术界已开展了一系列研究，包括信号处理设计^[8-10]以及在边缘智能中的应用^[11-12]。然而，现有关于通信-感知-计算一体化的文献尚未考虑边缘服务器和终端设备的算力配置。实际上，感知、通信和计算过程都依赖于算力配置，这一点在系统设计中至关重要。

针对上述问题，本文提出了一种新颖的、能适应边缘服务器和终端设备算力配置的通信-感知-计算一体化设计。终端设备可以感知数据、计算数据，并将数据卸载到边缘服务器进行后续处理。整个数据处理周期被划分为多个时隙，在这些时隙中可以执行数据感知、卸载和计算。为了实现高效的数据处理，本文对每个时隙中的数据感知速率、卸载量和卸载速率进行了联合优化。

1 系统模型

通信-感知-计算一体化系统设计如图1所示，该系统由一个具备数据感知、通信和计算能力的终端设备以及一个协助数据计算的服务器组成。在该系统中，数据处理任务要求终端设备采集 L bit 的数据，并在截止时间 T 前完成计算。由于其计算能力有限，终端设备会自适应地将部分数据卸载到服务器进行协同计算。服务器以恒定的处理频率 f_h 运行，但会间歇性地处理其他计算

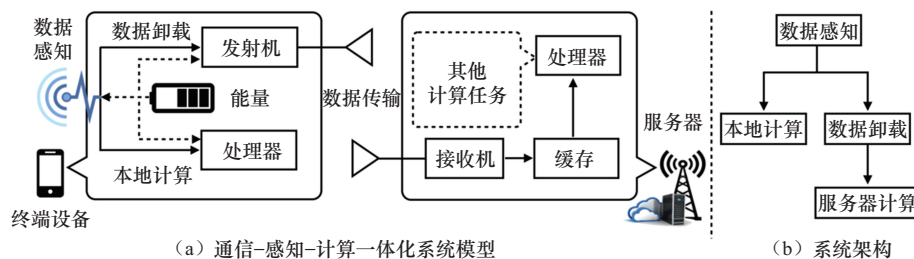


图1 通信-感知-计算一体化系统设计

任务。当有其他计算任务时，服务器的处理状态被识别为繁忙；否则，其处理状态被视为空闲。服务器只有在处理状态为空闲时，才能计算由终端设备卸载的数据。

根据文献[12]的方法，服务器的算力配置被定义为服务器直到时间点 $t \in [0, T]$ 累计可以计算的数据量，记为 $U_{\text{bit}}(t)$ 。假设存在 K 个时间点，记为 $\{t_1, \dots, t_k, \dots, t_K\}$ ，在这些时间点上，服务器的处理状态在空闲和繁忙之间切换。因此，总时间 T 被划分为 K 个时段，每个时段持续时间为 $\tau_k = t_k - t_{k-1}$ ，其中 $t_0 = 0$ 且 $t_K = T$ 。假设终端设备已知服务器的处理状态，对于时段 τ_k ，其处理状态由 $a_k \in \{0, 1\}$ 表示。当处理状态为空闲时， $a_k = 1$ ，否则 $a_k = 0$ 。设 C_h 表示在服务器上计算 1 bit 数据所需的处理周期数。在时段 τ_k 内，服务器可计算的最大数据量为 $d_{k, \max} = (a_k \tau_k f_h) / C_h$ 。直到时间 t_k ，服务器可

计算的最大累计数据量为 $U_{\text{bit}, k} = \sum_{j=1}^k (a_j \tau_j f_h) / C_h$ 。

为实现高能效的数据处理，本文对数据感知、通信和计算过程进行联合设计，旨在以最低的能耗完成任务。

1.1 数据感知模型

本文用 L_k 表示终端设备在时段 τ_k 内感知的数据量。由于在每个时段内采用恒定的速率进行感知被证明是高能效的^[13]，因此时段 τ_k 内的感知速率 s_k 是固定的，满足 $s_k = L_k / \tau_k$ 。为确保终端设备能在截止时间 T 前感知到 L bit 的数据，需要满足 $\sum_{k=1}^K s_k \tau_k = L$ 。根据文献[13]，时段 τ_k 内的感知功耗为 $P_{s, k} = \alpha C_s^2 s_k^2$ ，其中 C_s 是感知 1 bit 数据所需的处理周期数，而 α 是一个由电路决定的常数。因此，在整个时间范围内用于数据感知的总能耗为 $E_s = \sum_{k=1}^K \alpha C_s^2 s_k^2 \tau_k$ 。

1.2 数据卸载模型

本文用 l_k 表示在时段 τ_k 内从终端设备卸载

到服务器的数据量。由于在每个时段内采用恒定的速率进行传输被证明是高能效的^[14]，因此时段 τ_k 内的卸载速率 r_k 是固定的，满足 $r_k = l_k / \tau_k$ 。根据香农容量公式，时段 τ_k 的卸载功耗为 $P_{(o, k)} = (\sigma^2 / g) (e^{\frac{r_k}{B}} - 1)$ ，其中 σ^2 、 g 和 B 分别表示噪声功率、有效信道功率增益和频谱带宽。因此，在整个时间范围内用于数据卸载的总能耗为 $E_o = \sum_{k=1}^K \left(\frac{\sigma^2}{g} \right) \left(e^{\frac{r_k}{B}} - 1 \right) \tau_k$ 。

1.3 数据计算模型

本文用 θ 表示卸载到服务器的数据比例，意味着总量为 $l = \theta L$ 的数据将被卸载到服务器进行计算，而剩余的总量为 $(1 - \theta)L$ 的数据将在终端设备上本地计算。

1.3.1 本地计算模型

考虑截止时间约束，本地计算应满足：

$$\frac{C_l(1-\theta)L}{f_l} \leq T \quad (1)$$

其中， f_l 为终端设备的恒定处理频率， C_l 为在终端设备上计算 1 bit 数据所需的处理周期数。由于卸载到服务器的数据比例不能低于 0，因此最小

卸载比例为 $\theta_{\min} = \left[1 - \frac{Tf_l}{LC_l} \right]^+$ 。根据文献[14]，每个处理周期的能耗为 $P_{\text{cyc}} = \gamma f_l^2$ ，其中 γ 为由开关电容决定的常数。因此，本地计算的总能耗为 $E_l = C_l(1 - \theta)LP_{\text{cyc}}$ 。由于数据在被终端设备感知前无法被卸载或计算，在时段 τ_k 存在如下因果约束：

$$\sum_{j=1}^k s_j \tau_j \geq \sum_{j=1}^k r_j \tau_j + D_{k, \min}, k = 1, 2, \dots, K \quad (2)$$

其中， $D_{k, \min} = \left[\sum_{j=1}^k \tau_j - \left(T - \frac{(1-\theta)LC_l}{f_l} \right) \right]^+ f_l / C_l$ 为直到时段 τ_k 需要本地计算的最小数据量。

1.3.2 边缘计算模型

对于边缘服务器上的计算，当前可用的算力



不能用于处理未来数据，这被称为处理器的实时性约束。本文用 d_k 表示服务器在时段 τ_k 计算的数据量，用 B_k 表示在时段 τ_k 结束时，已卸载到服务器但尚未被计算的剩余数据量。因此， B_k 和 d_k 可分别表示为：

$$B_k = \sum_{j=1}^k l_j - \sum_{j=1}^k d_j \quad (3)$$

$$d_k = \min \left\{ B_{k-1} + l_k, \frac{a_k \tau_k f_h}{C_h} \right\} \quad (4)$$

其中， $(B_{k-1} + l_k)$ 表示在时段 τ_k 内待处理的数据缓冲区大小。由于数据在卸载前无法在服务器上进行计算，因此存在一个卸载因果约束：

$$\sum_{j=1}^k r_j \tau_j \geq \sum_{j=1}^k d_j, k=1, 2, \dots, K \quad (5)$$

此外，考虑截止时间约束，边缘计算应满足：

$$\sum_{k=1}^K d_k = \sum_{k=1}^K l_k = l = \theta L \quad (6)$$

一个满足上述约束的可行感知速率和卸载速率集合如图2所示。所需的感知数据量和卸载数据量分别由灰色和黑色的阶梯状曲线的高度表示。感知速率和卸载速率则分别由 s_1, s_2 和 r_1, r_2 表示。

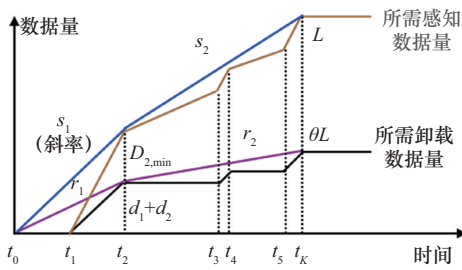


图2 一个满足约束的可行感知速率和卸载速率集合

2 高效的通信-感知-计算一体化设计

本文研究了用于自适应数据感知、卸载和计算的高效通信-感知-计算一体化设计，通过联合优化各时段的数据卸载比例以及感知和卸载速率，以最小化终端设备完成数据处理任务的总能

耗。相应的优化问题构建如下。

$$(P1) \min_{\theta, \{s_k\}, \{r_k\}} C_l(1-\theta)LP_{cyc} + \sum_{k=1}^K \left[\alpha C_h^2 s_k^2 + \left(\frac{\sigma^2}{g} \right) \left(e^{\frac{r_k}{B}} - 1 \right) \right] \tau_k \quad (7)$$

$$\text{s.t. } \theta_{\min} \leq \theta \leq 1 \quad (8)$$

$$\sum_{k=1}^K s_k \tau_k = L \quad (9)$$

$$\sum_{k=1}^K d_k = \sum_{k=1}^K r_k \tau_k = \theta L \quad (10)$$

$$\sum_{j=1}^k s_j \tau_j \geq \sum_{j=1}^k r_j \tau_j + D_{k,\min}, k=1, 2, \dots, K \quad (11)$$

$$\sum_{j=1}^k r_j \tau_j \geq \sum_{j=1}^k d_j, k=1, 2, \dots, K \quad (12)$$

其中，式(8)表示可行的卸载数据比例范围，式(9)表示数据感知的截止时间约束，式(10)表示数据卸载和计算的截止时间约束，式(11)表示数据感知的因果约束，式(12)表示数据卸载的因果约束。优化变量之间的耦合关系使得问题P1的求解并非易事。然而，可以观察到，当卸载的数据量不大于服务器可计算的最大数据量，即 $l \leq U_{\text{bit},K}$ 时，问题P1存在可行解。为了简化流程，考虑在服务器算力被完全利用的条件下求解问题P1，即 $l = U_{\text{bit},K}$ 。在这种情况下，可行解需要利用服务器所有的处理空闲时段来满足截止时间约束。因此，可以得到 $d_k = \frac{a_k \tau_k f_h}{C_h} \leq B_{k-1} + l_k$,

$k=1, 2, \dots, K$ 。根据式(10)，可以得到 $\theta = \frac{\sum_{k=1}^K a_k \tau_k f_h}{C_h L}$ 。值得注意的是，必须满足 $\theta_{\min} \leq$

$\sum_{k=1}^K \frac{a_k \tau_k f_h}{C_h L} \leq 1$ 。相应的问题可以简化为：

$$(P2) \min_{\{s_k\}, \{r_k\}} C_l(1-\theta)LP_{cyc} + \sum_{k=1}^K \left[\alpha C_h^2 s_k^2 + \left(\frac{\sigma^2}{g} \right) \left(e^{\frac{r_k}{B}} - 1 \right) \right] \tau_k \quad (13)$$

$$\text{s.t. } \sum_{k=1}^K s_k \tau_k = L \quad (14)$$

$$\sum_{k=1}^K r_k \tau_k = \theta L \quad (15)$$

$$\sum_{j=1}^k s_j \tau_j \geq \sum_{j=1}^k r_j \tau_j + D_{k,\min}, k=1, 2, \dots, K \quad (16)$$

$$\sum_{j=1}^k r_j \tau_j \geq \sum_{j=1}^k \frac{a_j \tau_j f_h}{C_h}, k=1, 2, \dots, K \quad (17)$$

然而，在约束 (16) 中，优化变量仍然是耦合的。为解决此问题，首先通过求解以下子问题来优化 $\{r_k\}$ ：

$$(P3) \min_{\{r_k\}} C_l(1-\theta)LP_{\text{cyc}} + \sum_{k=1}^K \left(\frac{\sigma^2}{g} \right) \left(e^{\frac{r_k}{B}} - 1 \right) \tau_k \quad (18)$$

$$\text{s.t.} \sum_{k=1}^K r_k \tau_k = \theta L \quad (19)$$

$$\sum_{j=1}^k r_j \tau_j \geq \sum_{j=1}^k \frac{a_j \tau_j f_h}{C_h}, k=1, 2, \dots, K \quad (20)$$

需要注意的是，卸载速率的优化目标是在由处理器配置 $\{(a_k \tau_k f_h)/C_h\}$ 决定的“地板”上方，构建一条从起点（如图 2 中的 $(t_0, 0)$ ）到终点（如图 2 中的 $(t_K, \theta L)$ ）的最短路径。这可以通过“从起点到终点，在地板上方拉紧一根线”来完成，即著名的拉线算法。在算力完全利用情况下，最优卸载速率应满足以下性质。

引理 1 （算力完全利用情况下的最优卸载速率）最优卸载速率在各个时段上是单调非递增的，即 $r_k \geq r_{k+1}, k=1, 2, \dots, K$ 。此外，卸载速率的下降只可能发生在卸载因果约束被激活（即取等号）时，即仅当 $\sum_{j=1}^k r_j \tau_j = \sum_{j=1}^k (a_k \tau_k f_h)/C_h$ 时，才可能有 $r_k > r_{k+1}$ 。

接着通过求解以下子问题来优化 $\{s_k\}$ ：

$$(P4) \min_{\{s_k\}} \sum_{k=1}^K \alpha C_h^2 s_k^2 \tau_k \quad (21)$$

$$\text{s.t.} \sum_{k=1}^K s_k \tau_k = L \quad (22)$$

$$\sum_{j=1}^k s_j \tau_j \geq \sum_{j=1}^k \frac{a_j \tau_j f_h}{C_h} + D_{k,\min}, k=1, 2, \dots, K \quad (23)$$

感知速率的优化目标是在由 $(a_j \tau_j f_h)/C_h + D_{k,\min}$ 决定的“地板”上方，构建一条从起点（如图 2 中的 $(t_0, 0)$ ）到终点（如图 2 中的 (t_K, L) ）的最短路径，这也具有拉线结构。在算力完全利用的情况下，最优感知速率应满足以下性质。

引理 2 （算力完全利用情况下的最优感知速率）最优感知速率在各个时段上是单调非递增的，即 $s_k \geq s_{k+1}, k=1, 2, \dots, K$ 。此外，感知速率的下降只可能发生在感知因果约束被激活时，即仅当 $\sum_{j=1}^k s_j \tau_j = \sum_{j=1}^k (a_k \tau_k f_h)/C_h + D_{k,\min}$ 时，才可能有 $s_k > s_{k+1}$ 。

基于引理 1 和引理 2 给出的速率单调非递增性质，可以应用拉线算法来获取最优解。对于问题 P3，通过设置“地板”值 $\{F_k = \sum_{j=1}^k (a_j \tau_j f_h)/C_h\}$ 得到最优卸载速率；对于问题 P4，通过设置“地板”值 $\{F_k = (a_j \tau_j f_h)/C_h + D_{k,\min}\}$ 得到最优感知速率。该算法通过迭代寻找具有最大斜率的速率曲线来实现，最优速率控制的拉线算法如算法 1 所示。问题 P2 的最优感知速率和卸载速率由命题 1 给出。

算法 1 最优速率控制的拉线算法

输入 时间点 $\{t_k\}$ 处的“地板”值 $\{F_k\}$

输出 最优速率 $\{R_i^*\}$ 和对应的持续时间 $\{T_i^*\}$

初始化 $j_0=0, i=0$

当 $j_i < K$

更新 $i \leftarrow i+1$

计算 $j_i = \operatorname{argmax}_{j: j_{i-1} < j \leq K} \left\{ \frac{F_j - F_{j_{i-1}}}{t_j - t_{j_{i-1}}} \right\}$

计算 $R_i^* = \left\{ \frac{F_j - F_{j_{i-1}}}{t_j - t_{j_{i-1}}} \right\}$

计算 $T_i^* = t_j - t_{j_{i-1}}$

结束循环

返回最优速率 $\{R_i^*\}$ 和对应的持续时间 $\{T_i^*\}$

命题 1 （算力完全利用情况下的最优感知



速率和卸载速率)通过算法1为问题P3和P4分别求得的最优卸载速率 $\{r_k^*\}$ 和最优感知速率 $\{s_k^*\}$,同样是问题P2的最优解。

证明:由于问题P2的目标函数是问题P3和P4目标函数之和,如果 $\{r_k^*\}$ 和 $\{s_k^*\}$ 对问题P2是可行的,那么它们就是最优解。可以观察到,唯一导致感知速率和卸载速率耦合的约束是式(16)。因此,只要 $\{r_k^*\}$ 和 $\{s_k^*\}$ 满足式(16),它们对于问题P2就是可行的。可以通过反证法证明,假设约束(16)在某个时段 τ_k 被违反,即 $s_j^* \tau_j < r_j^* \tau_j + D_{k,\min}$,此时可以将卸载速率降低至 r'_k ,使得

$r'_k \tau_k = \sum_{j=1}^k s_j^* \tau_j - D_{k,\min} - \sum_{j=1}^{k-1} r_j^* \tau_j$ 。由于 $\sum_{j=1}^k s_j^* \tau_j \geq \sum_{j=1}^k (a_j \tau_j f_h) / C_h + D_{k,\min}$ 对于问题P4成立,可以得到

$$r'_k \tau_k + \sum_{j=1}^{k-1} r_j^* \tau_j \geq \sum_{j=1}^k (a_j \tau_j f_h) / C_h, \text{ 这表明 } r'_k \text{ 对于问题P3是可行的。}$$

因此,可以通过均衡速率 r_k 和 r_{k+1} 来降低能耗,这与 r_k^* 的最优性相矛盾,故假设不成立。

3 数值结果与分析

本文提出的算法通过仿真进行验证。相关参数设置如下:服务器的处理频率 f_h 设为5 000 Hz,终端设备的本地处理频率 f_l 设为1 000 Hz,在终端设备上感知或在终端设备/服务器上计算每比特数据所需的处理周期数均为 $C_s = C_h = C_l = 500$ 次/bit,由电路决定的常数 α 设为 10^{-15} ,噪声功率 σ^2 设为0,有效信道功率增益 h 设为1,频谱带宽 B 设为1 MHz。根据文献[13],每个处理周期的能耗为 $P_{\text{cyc}} = 10^{-10}$ J。

服务器算力被完全利用情况下的最优感知速率和卸载速率如图3所示。服务器的算力在时间段 $[0, 1]$ s、 $[3, 5]$ s和 $[9, 10]$ s内为空闲。由图3可以观察到,最优的感知速率和卸载速率控制满足拉线结构,这与理论分析一致。

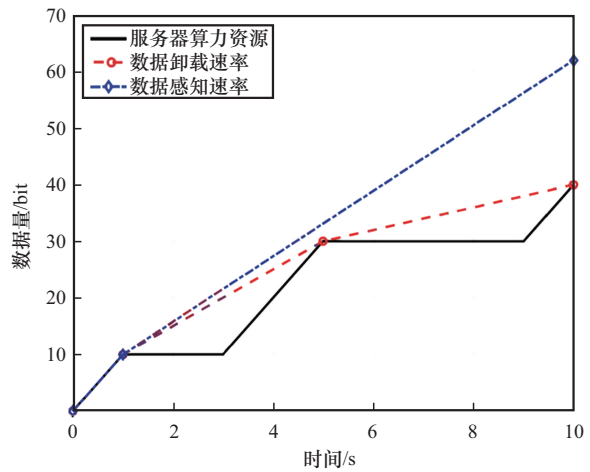


图3 服务器算力被完全利用情况下的最优感知速率和卸载速率

服务器处理器功率被完全利用情况下的能耗与总数据量的关系如图4所示。由图4可知,总能耗随数据量的增加而增加,这是因为感知、卸载和计算更多的数据会导致更高的能量消耗。此外,本文引入了3种采用随机但可行的感知和/或卸载速率的基准方案作为对比。本文所提出的方案通过联合优化感知与卸载速率,其性能优于其他基准方案。在基准方案中,仅优化感知速率(卸载速率随机设置)的方案性能优于仅优化卸载速率(感知速率随机设置)的方案,而感知与卸载速率均随机设置的方案性能最差。

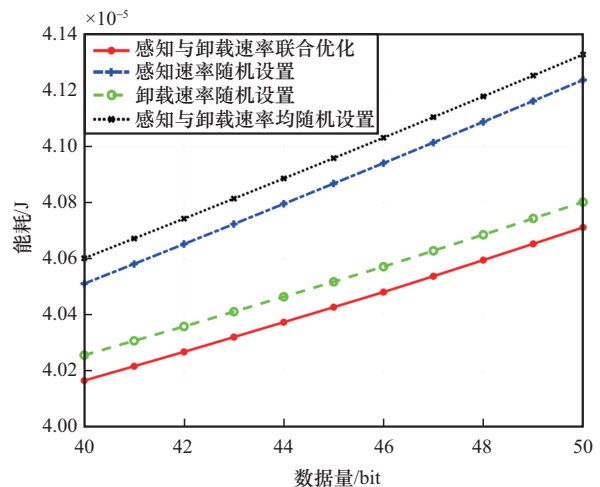


图4 服务器处理器功率被完全利用情况下的能耗与总数据量的关系

数据卸载比例对系统能耗的影响如图5所示。由图5可知,总能耗随卸载比例的增加先减少后增加,这表明存在一个最小化能耗的最优卸载比例。此外,能耗与卸载比例的关系在不同的终端设备和服务器处理频率设置下会发生变化,这表明最优卸载比例依据终端设备和服务器处理频率的不同设置而变化。当终端设备处理频率较大时,节能的方案倾向于在本地计算更多数据;当服务器处理频率较大时,更多数据倾向于卸载到服务器进行计算。

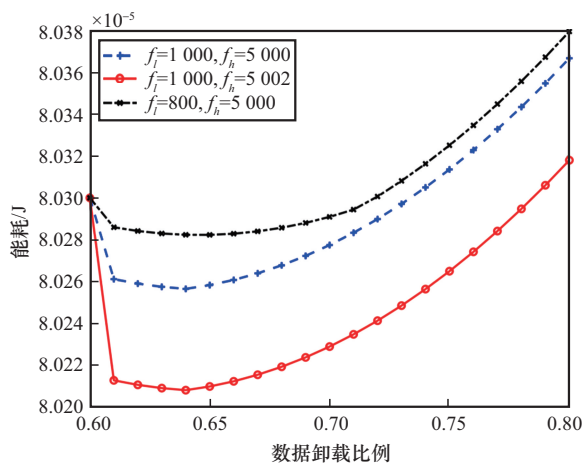


图5 数据卸载比例对系统能耗的影响

4 结束语

为了在物联网中实现高效的数据处理,本文提出了一种通信-感知-计算一体化设计。该设计根据终端设备和服务器的处理器配置,对数据感知速率、卸载量和卸载速率进行联合优化。为求解由此产生的复杂优化问题,考虑了服务器算力被完全利用的情况。在这种情况下,数据卸载率直接由服务器的算力配置决定,而最优的感知与卸载速率可以通过拉线算法获得。这项工作为通信-感知-计算一体化这一新兴研究领域做出了贡献,未来有多个研究问题值得进一步探讨,如设备调度和多服务器间的协作。

参考文献:

- [1] Al-Fuqaha A, Guizani M, Mohammadi M, et al. Internet of things: a survey on enabling technologies, protocols, and applications[J]. IEEE Communications Surveys & Tutorials, 2015, 17(4): 2347-2376.
- [2] Mao Y Y, You C S, Zhang J, et al. A survey on mobile edge computing: the communication perspective[J]. IEEE Communications Surveys & Tutorials, 2017, 19(4): 2322-2358.
- [3] Abbas N, Zhang Y, Taherkordi A, et al. Mobile edge computing: a survey[J]. IEEE Internet of Things Journal, 2018, 5(1): 450-465.
- [4] Mach P, Becvar Z. Mobile edge computing: a survey on architecture and computation offloading[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628-1656.
- [5] Cui Y H, Liu F, Jing X J, et al. Integrating sensing and communications for ubiquitous IoT: applications, trends, and challenges[J]. IEEE Network, 2021, 35(5): 158-167.
- [6] Liu A, Huang Z, Li M, et al. A survey on fundamental limits of integrated sensing and communication[J]. IEEE Communications Surveys & Tutorials, 2022, 24(2): 994-1034.
- [7] Wen D Z, Zhou Y, Li X Y, et al. A survey on integrated sensing, communication, and computation[PP]. arXiv preprint, 2024: 2408.08074.
- [8] Li X Y, Gong Y, Huang K B, et al. Over-the-air integrated sensing, communication, and computation in IoT networks[J]. IEEE Wireless Communications, 2023, 30(1): 32-38.
- [9] Li X Y, Liu F, Zhou Z Q, et al. Integrated sensing, communication, and computation over-the-air: MIMO beamforming design[J]. IEEE Transactions on Wireless Communications, 2023, 22(8): 5383-5398.
- [10] Wang S, Gong Y, Li X Y, et al. Integrated sensing, communication, and computation over the air: beam pattern design for wireless sensor networks[J]. IEEE Internet of Things Journal, 2024, 11(6): 9681-9692.
- [11] Zhu G X, Lyu Z H, Jiao X, et al. Pushing AI to wireless network edge: an overview on integrated sensing, communication, and computation towards 6G[J]. Science China Information Sciences, 2023, 66(3): 130301.
- [12] Wen D Z, Li X Y, Zhou Y, et al. Integrated sensing-communication-computation for edge artificial intelligence[J]. IEEE Internet of Things Magazine, 2024, 7(4): 14-20.
- [13] You C S, Huang K B. Exploiting non-causal CPU-state information for energy-efficient mobile cooperative computing[J]. IEEE



Transactions on Wireless Communications, 2018, 17(6): 4104-4117.

- [14] Zhou Z Q, Li X Y, You C S, et al. Joint sensing and communication-rate control for energy efficient mobile crowd sensing[J]. IEEE Transactions on Wireless Communications, 2023, 22(2): 1314-1327.

[作者简介]



周梓钦（1998-），男，南方科技大学工学院访问学生，主要研究方向为通信感知一体化。



周尚语（2005-），男，南方科技大学工学院在读，主要研究方向为智能物联网。



韩子栋（1990-），男，博士，南方科技大学工学院高级研究学者、高级工程师，主要研究方向为工业物联网、无线控制、边缘智能等。



曹晓雯（1995-），女，博士，深圳大学电子与信息工程学院助理教授、硕士生导师，主要研究方向为边缘智能、空中计算、可移动天线等。



李晓阳（1994-），男，博士，南方科技大学工学院助理教授、博士生导师，主要研究方向为通信感知计算一体化、智能物联网等。