1 系统模型

1.1 网络模型与时间结构

为便于描述，定义车辆集合、路侧单元（RSU）集合以及时隙集合分别为：

$$\mathcal{I} = \{1, 2, \ldots, I\}, \tag{1}$$

$$\mathcal{N} = \{1, 2, \ldots, N\}, \tag{2}$$

$$\mathcal{T} = \{1, 2, \ldots, T\}. \tag{3}$$

其中，$\mathcal{I}$ 表示系统中的车辆集合，$|\mathcal{I}| = I$；$\mathcal{N}$ 表示边缘节点集合（仅包含RSU），$|\mathcal{N}| = N$；$\mathcal{T}$ 表示离散时隙集合。本文采用分时隙建模，假设同一时隙内网络状态近似不变，不同时隙之间可能发生变化。

1.2 任务模型

1.2.1 任务到达

在每个时隙 t，车辆 i 生成的新任务数服从参数为 λ 的泊松分布：

$$M_i(t) \sim \text{Poisson}(\lambda), \tag{4}$$

车辆 i 在时隙 t 生成的任务索引集合定义为：

$$\mathcal{M}_i(t) = \{1, 2, \ldots, M_i(t)\}. \tag{5}$$

式 (4) 刻画了车载任务到达的随机性；λ 越大，系统任务负载越高，队列更易拥塞。

1.2.2 单任务三元组与取值范围

车辆 i 在时隙 t 生成的第 k 个任务定义为三元组：

$$\mathcal{T}_{i,k}(t) = \big(D_{i,k}(t),\, C_{i,k}(t),\, \tau_{i,k}(t)\big), \tag{6}$$

其中 $D_{i,k}(t)$ 表示任务数据量（bit），$C_{i,k}(t)$ 表示任务所需 CPU 周期数（cycles），$\tau_{i,k}(t)$ 表示任务最大允许完成时间（截止时间，s）。为反映任务属性的波动性，给出如下范围约束：

$$D_i^{\min} \le D_{i,k}(t) \le D_i^{\max}, \tag{7}$$

$$C_i^{\min} \le C_{i,k}(t) \le C_i^{\max}, \tag{8}$$

$$\tau_i^{\min} \le \tau_{i,k}(t) \le \tau_i^{\max}. \tag{9}$$

式 (7)–(9) 分别给出任务数据量、计算量和截止时间的上下界。

1.3 任务划分模型

任务可被划分为本地处理子任务与卸载处理子任务。令任务划分比为 $\alpha_{i,k}(t) \in [0,1]$：

$$\alpha_{i,k}(t) \in [0,1], \tag{10}$$

则划分后的本地/卸载子任务数据量满足：

$$D_{i,k}^{\mathrm{loc}}(t) = \alpha_{i,k}(t) D_{i,k}(t), \quad D_{i,k}^{\mathrm{off}}(t) = \big(1-\alpha_{i,k}(t)\big) D_{i,k}(t), \tag{11}$$

划分后的本地/卸载子任务 CPU 周期满足：

$$C_{i,k}^{\mathrm{loc}}(t) = \alpha_{i,k}(t) C_{i,k}(t), \quad C_{i,k}^{\mathrm{off}}(t) = \big(1-\alpha_{i,k}(t)\big) C_{i,k}(t). \tag{12}$$

其中 $\alpha_{i,k}(t)=1$ 表示完全本地处理，$\alpha_{i,k}(t)=0$ 表示完全卸载处理。

1.4 通信模型）

车辆与 RSU、邻车之间通过无线链路进行卸载通信。发射功率视为常量参数：车辆 i 的 V2I 发射功率为 P_i^{V2I}，V2V 发射功率为 P_i^{V2V}。

1.4.1 V2I 上行速率（车辆 $\rightarrow$ RSU）

车辆 i 与 RSU j 之间的上行速率由 Shannon 公式给出：

$$R_{i,j}(t) = B_{i,j} \log_2\!\left(1 + \frac{P_i^{\mathrm{V2I}}\, h_{i,j}(t)}{\sigma^2}\right), \tag{13}$$

其中 $B_{i,j}$ 为链路带宽，$h_{i,j}(t)$ 为信道增益，σ^2 为噪声功率。为简化建模，信道增益采用路径损耗模型：

$$h_{i,j}(t) = G \cdot d_{i,j}(t)^{-\beta}, \tag{14}$$

其中 $d_{i,j}(t)$ 为车辆与 RSU 的距离，β 为路径损耗指数，G 为常数（含收发增益等）。

1.4.2 V2V 上行速率（车辆 → 邻车）

车辆 i 与邻车 u 的 V2V 速率为：

$$R_{i,u}^{\mathrm{V2V}}(t) = B_{i,u}^{\mathrm{V2V}} \log_2\left(1 + \frac{P_i^{\mathrm{V2V}}\, g_{i,u}(t)}{\sigma^2}\right), \tag{15}$$

其中 $g_{i,u}(t)$ 为 V2V 信道增益，同样用路径损耗表示：

$$g_{i,u}(t) = G_v \cdot d_{i,u}(t)^{-\beta_v}. \tag{16}$$

1.5 卸载决策变量

卸载部分可选择通过 V2I（卸载到 RSU）或 V2V（卸载到邻车）处理。定义二元卸载决策变量：

$$x_{i,k}^{\mathrm{V2I}}(t) \in \{0,1\}, \quad x_{i,k}^{\mathrm{V2V}}(t) \in \{0,1\}, \tag{17}$$

且满足卸载部分只能在一个地方处理：

$$x_{i,k}^{\mathrm{V2I}}(t) + x_{i,k}^{\mathrm{V2V}}(t) = 1. \tag{18}$$

当选择 V2I 时，需要指定目标 RSU $j_{i,k}(t) \in \mathcal{N}$；当选择 V2V 时，需要指定目标邻车 $u_{i,k}(t) \in \mathcal{U}_i(t)$（$\mathcal{U}_i(t)$ 为时隙 t 车辆 i 的候选邻车集合）。

1.6 任务优先级建模（Priority）

在多车辆任务卸载场景中，不同任务的紧迫性和重要性差异显著。为合理分配计算与通信资源并降低关键任务超时风险，本文在任务划分与调度过程中引入任务优先级。在任务划分模块中，高优先级任务倾向于保留更大比例在本地执行，以降低不确定的传输时延风险；在调度模块中，RSU 或邻车接收到多个待执行子任务时，可根据优先级进行队列排序，从而使高优先级任务优先获得传输与计算资源。

具体而言，每个任务的优先级由三个核心因素决定：时延代价、紧急度与任务关键度（任务类别）。

1.6.1 时延代价及其归一化

首先计算任务在不同候选执行方式下的估计完成时间：

$$\hat{T}_{i,k}^{\mathrm{loc}}(t), \quad \hat{T}_{i,k}^{\mathrm{V2I}}(t), \quad \hat{T}_{i,k}^{\mathrm{V2V}}(t). \tag{19}$$

其中 $\hat{T}_{i,k}^{\mathrm{loc}}(t)$ 为本地预计完成时延，$\hat{T}_{i,k}^{\mathrm{V2I}}(t)$ 为上传到某 RSU 的预计完成时延，$\hat{T}_{i,k}^{\mathrm{V2V}}(t)$ 为向邻车卸载的预计完成时延。为便于比较，将候选估计完成时间进行归一化（经验上下界由系统统计或经验设定）：

$$\tilde{T}_{i,k}(t) = \frac{\hat{T}_{i,k}(t) - T_{\min}}{T_{\max} - T_{\min}}. \tag{20}$$

其中 $\hat{T}_{i,k}(t)$ 表示在某一候选方式下的估计完成时延，$T_{\min}$ 与 $T_{\max}$ 分别为估计完成时间的下界与上界。

1.6.2 紧急度及其归一化

任务紧急性可由任务剩余可用时间来度量：

$$U_{i,k}(t) = \tau_{i,k}(t) - \hat{T}_{i,k}(t). \tag{21}$$

剩余时间越小，任务越紧急。对紧急度进行归一化处理：

$$\tilde{U}_{i,k}(t) = \frac{U_{i,k}(t) - U_{\min}}{U_{\max} - U_{\min}}. \tag{22}$$

其中 $U_{\min}$ 与 $U_{\max}$ 为经验上下界。

1.6.3 任务关键度（任务类别）

为体现不同任务类别的固有重要性，将任务类别映射为一个数值（任务关键度）：

$$\kappa_{i,k}(t). \tag{23}$$

其中 $\kappa_{i,k}(t)$ 越大表示任务固有重要性越高（例如安全相关任务通常高于娱乐类任务）。

1.6.4 优先级计算

将上述三个量线性组合得到任务优先级：

$$\pi_{i,k}(t) = w_1\tilde{T}_{i,k}(t) + w_2\tilde{U}_{i,k}(t) + w_3\kappa_{i,k}(t), \tag{24}$$

其中 w_1, w_2, w_3 为权重系数，用于平衡时延代价、紧急度与任务关键度对优先级的贡献。$\pi_{i,k}(t)$ 越大表示任务越重要，应在划分与调度中给予更高保障。

1.7 时延与能耗模型

为刻画队列等待效应，引入“队头任务未完成比例” $\rho(\cdot) \in [0,1]$，以反映队头任务可能尚未完全处理/传输的连续性。

1.7.1 本地方式（本地子任务）

车辆 i 的本地计算频率为 $f_i(t)$，取值范围为：

$$f_i^{\min} \le f_i(t) \le f_i^{\max}. \tag{25}$$

设本地计算队列为 $\mathcal{Q}_i^{\mathrm{loc}}(t)$，则本地子任务的等待时间为：

$$T_{i,k,\mathrm{loc}}^{\mathrm{wait}}(t) = \sum_{q \in \mathcal{Q}_i^{\mathrm{loc}}(t)} \frac{C_q}{f_i(t)} + \rho_i^{\mathrm{loc}}(t) \frac{C_{\mathrm{head}}(t)}{f_i(t)}. \tag{26}$$

本地计算时间为：

$$T_{i,k,\mathrm{loc}}^{\mathrm{exe}}(t) = \frac{C_{i,k}^{\mathrm{loc}}(t)}{f_i(t)}. \tag{27}$$

因此本地总时延为：

$$T_{i,k}^{\mathrm{loc}}(t) = T_{i,k,\mathrm{loc}}^{\mathrm{wait}}(t) + T_{i,k,\mathrm{loc}}^{\mathrm{exe}}(t). \tag{28}$$

本地计算能耗采用 DVFS 近似模型：

$$E_{i,k}^{\mathrm{loc}}(t) = \kappa\, f_i(t)^2\, C_{i,k}^{\mathrm{loc}}(t), \tag{29}$$

其中 κ 为能耗系数。

1.7.2 V2I 方式（卸载到 RSU）

当卸载子任务进入 V2I 调度队列 $\mathcal{Q}_i^{\mathrm{V2I}}(t)$ 时，其调度队列等待时间为：

$$T_{i,k,\mathrm{V2I-q}}^{\mathrm{wait}}(t) = \sum_{q \in \mathcal{Q}_i^{\mathrm{V2I}}(t)} T_{q,\mathrm{V2I}}^{\mathrm{tx}}(t) + \rho_i^{\mathrm{V2I}}(t)\, T_{\mathrm{head,V2I}}^{\mathrm{tx}}(t). \tag{30}$$

上传传输时间为：

$$T_{i,k,\mathrm{V2I}}^{\mathrm{tx}}(t) = \frac{D_{i,k}^{\mathrm{off}}(t)}{R_{i,j_{i,k}(t)}(t)}. \tag{31}$$

到达 RSU 后进入 RSU 计算队列 $\mathcal{Q}_j^{\mathrm{rsu}}(t)$，其计算等待时间为：

$$T_{i,k,\mathrm{rsu}}^{\mathrm{wait}}(t)=\sum_{q\in\mathcal{Q}_{j_{i,k}(t)}^{\mathrm{rsu}}(t)}\frac{C_q}{f_{j_{i,k}(t)}^{\mathrm{rsu}}}+\rho_{j_{i,k}(t)}^{\mathrm{rsu}}(t)\frac{C_{\mathrm{head}}(t)}{f_{j_{i,k}(t)}^{\mathrm{rsu}}}. \tag{32}$$

RSU 计算时间为：

$$T_{i,k,\mathrm{rsu}}^{\mathrm{exe}}(t)=\frac{C_{i,k}^{\mathrm{off}}(t)}{f_{j_{i,k}(t)}^{\mathrm{rsu}}}. \tag{33}$$

因此 V2I 总时延为：

$$T_{i,k}^{\mathrm{V2I}}(t)=T_{i,k,\mathrm{V2I-q}}^{\mathrm{wait}}(t)+T_{i,k,\mathrm{V2I}}^{\mathrm{tx}}(t)+T_{i,k,\mathrm{rsu}}^{\mathrm{wait}}(t)+T_{i,k,\mathrm{rsu}}^{\mathrm{exe}}(t). \tag{34}$$

V2I 能耗仅计车辆侧传输能耗（功率不优化，视为常量）：

$$E_{i,k}^{\mathrm{V2I}}(t)=P_i^{\mathrm{V2I}}\cdot T_{i,k,\mathrm{V2I}}^{\mathrm{tx}}(t). \tag{35}$$

1.7.3 V2V 方式（卸载到邻车）

卸载子任务进入 V2V 调度队列 $\mathcal{Q}_i^{\mathrm{V2V}}(t)$ 的等待时间为：

$$T_{i,k,\mathrm{V2V-q}}^{\mathrm{wait}}(t)=\sum_{q\in\mathcal{Q}_i^{\mathrm{V2V}}(t)}T_{q,\mathrm{V2V}}^{\mathrm{tx}}(t)+\rho_i^{\mathrm{V2V}}(t)\,T_{\mathrm{head,V2V}}^{\mathrm{tx}}(t). \tag{36}$$

V2V 传输时间为：

$$T_{i,k,\mathrm{V2V}}^{\mathrm{tx}}(t)=\frac{D_{i,k}^{\mathrm{off}}(t)}{R_{i,u_{i,k}(t)}^{\mathrm{V2V}}(t)}. \tag{37}$$

到达邻车后进入邻车计算队列 $\mathcal{Q}_u^{\mathrm{veh}}(t)$，其计算等待时间为：

$$T_{i,k,\mathrm{veh}}^{\mathrm{wait}}(t)=\sum_{q\in\mathcal{Q}_{u_{i,k}(t)}^{\mathrm{veh}}(t)}\frac{C_q}{f_{u_{i,k}(t)}(t)}+\rho_{u_{i,k}(t)}^{\mathrm{veh}}(t)\frac{C_{\mathrm{head}}(t)}{f_{u_{i,k}(t)}(t)}. \tag{38}$$

邻车计算时间为：

$$T_{i,k,\mathrm{veh}}^{\mathrm{exe}}(t)=\frac{C_{i,k}^{\mathrm{off}}(t)}{f_{u_{i,k}(t)}(t)}. \tag{39}$$

因此 V2V 总时延为：

$$T_{i,k}^{\mathrm{V2V}}(t)=T_{i,k,\mathrm{V2V-q}}^{\mathrm{wait}}(t)+T_{i,k,\mathrm{V2V}}^{\mathrm{tx}}(t)+T_{i,k,\mathrm{veh}}^{\mathrm{wait}}(t)+T_{i,k,\mathrm{veh}}^{\mathrm{exe}}(t). \tag{40}$$

V2V 能耗同样仅计车辆侧传输能耗：

$$E_{i,k}^{\mathrm{V2V}}(t)=P_i^{\mathrm{V2V}}\cdot T_{i,k,\mathrm{V2V}}^{\mathrm{tx}}(t). \tag{41}$$

1.7.4 任务总时延与总能耗

卸载部分时延与能耗为：

$$T_{i,k}^{\mathrm{off}}(t) = x_{i,k}^{\mathrm{V2I}}(t)\, T_{i,k}^{\mathrm{V2I}}(t) + x_{i,k}^{\mathrm{V2V}}(t)\, T_{i,k}^{\mathrm{V2V}}(t), \tag{42}$$

$$E_{i,k}^{\mathrm{off}}(t) = x_{i,k}^{\mathrm{V2I}}(t)\, E_{i,k}^{\mathrm{V2I}}(t) + x_{i,k}^{\mathrm{V2V}}(t)\, E_{i,k}^{\mathrm{V2V}}(t). \tag{43}$$

任务总时延与总能耗分别为：

$$T_{i,k}(t) = T_{i,k}^{\mathrm{loc}}(t) + T_{i,k}^{\mathrm{off}}(t), \tag{44}$$

$$E_{i,k}(t) = E_{i,k}^{\mathrm{loc}}(t) + E_{i,k}^{\mathrm{off}}(t). \tag{45}$$

2 问题描述

2.1 超时惩罚（Penalty）

若任务总时延超过其截止时间，则产生超时惩罚。定义超时指示函数为：

$$\mathbf{1}_{i,k}(t) = \begin{cases} 1, & T_{i,k}(t) > \tau_{i,k}(t), \\ 0, & T_{i,k}(t) \le \tau_{i,k}(t). \end{cases} \tag{46}$$

定义超时量为：

$$\Delta_{i,k}(t) = \big[T_{i,k}(t) - \tau_{i,k}(t)\big]^{+}. \tag{47}$$

超时惩罚受任务优先级与紧急度的权重影响。给出惩罚形式：

$$\Phi_{i,k}(t) = \mathbf{1}_{i,k}(t) \cdot \Big(\eta_1\, \pi_{i,k}(t) + \eta_2\, \tilde{U}_{i,k}(t)\Big), \tag{48}$$

其中 η_1, η_2 为优先级与紧急度对惩罚的影响权重，$\pi_{i,k}(t)$ 为任务优先级，$\tilde{U}_{i,k}(t)$ 为归一化紧急度。

若惩罚随超时量增长，则公式为：

$$\Phi_{i,k}(t) = \Delta_{i,k}(t) \cdot \Big(\eta_1\, \pi_{i,k}(t) + \eta_2\, \tilde{U}_{i,k}(t)\Big). \tag{49}$$

2.2 单任务成本函数

定义单任务总成本函数为：

$$J_{i,k}(t) = \lambda_1 T_{i,k}(t) + \lambda_2 E_{i,k}(t) + \lambda_3 \Phi_{i,k}(t), \tag{50}$$

其中 $\lambda_1, \lambda_2, \lambda_3$ 为权重系数，用于平衡时延、能耗与超时惩罚。

2.3 优化目标与约束

最终优化目标是在整个时间周期内最小化所有任务的期望系统成本：

$$\min_{\{\alpha,f,x,j,u\}} \mathbb{E}\left[\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{M}_i(t)} J_{i,k}(t)\right] \tag{51}$$

$$\text{s.t.} \quad 0 \le \alpha_{i,k}(t) \le 1, \quad \forall i \in \mathcal{I},\, k \in \mathcal{M}_i(t),\, t \in \mathcal{T}, \tag{52}$$

$$f_i^{\min} \le f_i(t) \le f_i^{\max}, \quad \forall i \in \mathcal{I},\, t \in \mathcal{T}, \tag{53}$$

$$x_{i,k}^{\mathrm{V2I}}(t) \in \{0,1\},\ x_{i,k}^{\mathrm{V2V}}(t) \in \{0,1\}, \quad \forall i,k,t, \tag{54}$$

$$x_{i,k}^{\mathrm{V2I}}(t) + x_{i,k}^{\mathrm{V2V}}(t) = 1, \quad \forall i,k,t, \tag{55}$$

$$j_{i,k}(t) \in \mathcal{N} \quad \text{当} x_{i,k}^{\mathrm{V2I}}(t) = 1,\ \forall i,k,t, \tag{56}$$

$$u_{i,k}(t) \in \mathcal{U}_i(t) \quad \text{当} x_{i,k}^{\mathrm{V2V}}(t) = 1,\ \forall i,k,t. \tag{57}$$