边缘计算环境下网络切片的动态调度策略研究

宋 莉[1]
SONG Li

摘 要 针对边缘计算场景下，资源异构与业务动态变化引发的网络切片调度不稳定问题，文章构建了一套面向多类 SLA 业务的动态调度策略框架，提出一种结合优先级驱动与深度强化学习的多策略调度算法。通过构建资源约束矩阵、设计服务优先级评分函数与多目标优化模型，并引入轻量级分布式控制逻辑，仿真实验验证：所提方法能够有效削减平均响应时延与节点迁移次数，提升资源利用率和服务可用性，同时具备优异的部署适配能力与调度鲁棒性。

关键词 边缘计算；网络切片；动态调度；服务质量（QoS）；深度强化学习

doi：10.3969/j.issn.1672-9528.2026.06.023

0 引言

随着边缘计算在智能制造、智慧园区等场景的广泛部署，网络切片面临跨节点资源动态调度与多业务差异化保障的双重挑战。传统静态映射或贪婪调度策略在复杂边缘环境中难以兼顾低延迟与资源高效利用。本文聚焦异构边缘架构下的多租户业务，探索融合规则驱动与学习增强策略的调度机制，构建从建模、优化到算法实现的完整路径。

1. 兰州文理学院 甘肃兰州 730000

1 网络切片与边缘计算协同需求分析

1.1 网络切片资源模型与边缘系统结构

网络切片在 RAN-Core-Transport 架构中构建虚拟化资源隔离环境，控制面部署 AMF、SMF 等功能实体，数据面基

4 结论与展望

本文提出了一种基于改进残差网络的终端设备状态智能诊断与 AI 辅助的故障修复方法，在残差网络的基础上引入了 LSTM 和注意力机制，增强对时序数据的特征提取能力。

在改进残差网络基础上设计的 AI 辅助故障修复系统，可以做到发现故障、分析原因到自动修复，能大幅提升运维效率。

实验结果显示，该方法诊断准确率达到了 98.2%，修复成功率达到了 91.3%，均优于传统的人工运维。应用案例进一步证明了本文方法的实用性。未来的研究可以进一步提升模型的通用性，让智能运维技术的应用场景更广泛。

参考文献：

[1] 中华人民共和国工业和信息化部 . 2025 年前 7 个月通信业经济运行情况 [R]. 北京 : 工业和信息化部 ,2025.

[2] 刘文婷 , 徐昶 , 朱茂中 , 等 . 基于 AI Agent 技术实现宽带故障的智能诊断应用研究 [J]. 电信工程技术与标准化 , 2025, 38(5): 30-36.

[3] 谢道祥 . 智能诊断与故障预判技术在广播电视无线发射台站的应用研究 [J]. 广播与电视技术 ,2025,52(7):80-83.

[4] Wu Keyu, Yu Qianjin, Mei Manlin ,et al. TN-AutoRCA: benchmark construction and agentic framework for self-improving alarm-based root cause analysis in telecommunication networks[PP/OL].(2025-07-28)[2026-05-23].https://doi.org/10.48550/arXiv.2507.18190.

[5] 谭瑛 , 黄彬 , 关俊波 , 等 . 基于 5G 消息和深度学习的宽带故障排查方案研究 [J]. 广西通信技术 ,2022(3):26-29.

[6] 唐嘉辉 , 成小乐 , 孙戬 , 等 . 基于注意力残差神经网络的滚动轴承复合故障诊断方法 [J]. 机械设计 ,2025,42(11):99-105.

[7]He Kaiming, Zhang Xiangyu, Ren Shaoqing,et al. Deep Residual Learning for Image Recognition [C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2016:770-778.

【作者简介】

张子樵（1979—），男，湖北武汉人，本科，高级工程师、高级专家，研究方向：互联网技术、终端与业务、人工智能。

（收稿日期：2025-11-21 修回日期：2026-06-02）

于 UPF 实现路径分离。每条切片作为独立 NSI 实例运行，绑定特定 VNF 集合与物理资源映射关系，支持动态调度与服务级别协议（SLA）差异化保障。

边缘系统采用“中心云—边缘节点—终端”三层结构，节点配置异构算力资源（如 ARM 2.4 GHz/4 核 CPU、集成 GPU、8 GB 内存、128 GB SSD），运行 Kubernetes 或 K3s 调度框架。资源波动性强，需实时感知与调控。资源建模维度包括：CPU 利用率（%）、GPU 占用率、可分配时隙（ms）；I/O 带宽（MB/s）、剩余存储容量（GB）；链路带宽（Mbit/s）、端到端时延（ms）、丢包率（%）。系统调度需实现切片资源需求向量与边缘节点能力向量的动态映射，保证调度可行性与可迁移性，即：迁移延迟 <50 ms，服务中断 ≤10 ms。

1.2 调度约束条件与策略设计目标

本研究以某工业园区边缘计算平台为研究场景，平台由 6 个边缘节点与 1 个中心调度控制器组成，节点间通过 10 Gbit/s 以太网互联，平均链路时延 3.2 ms。

各节点采用 ARM A78 2.4 GHz CPU、512 CUDA Cores GPU 与 16 GB DDR4 内存构成异构计算集群，单节点可提供约 2.1 TFLOPS 算力。平台运行 Kubernetes 容器编排系统，切片实例通过虚拟化网络功能（VNF）部署，形成可动态调度的多租户资源池。园区内业务类型涵盖高清视频监控、机器视觉检测与协同机器人控制三类，其中 eMBB 业务带宽需求 120 Mbit/s、时延敏感度低；URLLC 任务平均报文周期 10 ms、时延阈值 ≤ 20 ms；mMTC 任务连接密度 $>10^6$ device/km²、上行带宽需求波动范围 0.5~2 Mbit/s。平台资源利用率受业务周期性负载波动影响显著，峰值 CPU 占用率达 92%，链路负载变化幅度约 ±25%。在高并发状态下，多切片服务竞争计算与通信资源，导致时延抖动与资源抢占频繁发生。不同节点间的算力差异（最大差异比 1:1.8）与任务迁移路径的延迟特性（平均迁移开销 38 ms）形成调度主要约束[1]。系统需在异构节点环境中同时维持服务等级协议（SLA）约束、任务连续性及资源分配稳定性，以支撑园区级低时延、高可靠的业务运行环境。

2 网络切片动态调度模型设计

2.1 调度系统建模与触发机制构建

调度系统以离散事件驱动模型构建，状态空间 S_t 表示所有边缘节点在时刻 t 的资源占用状态，覆盖计算资源 R_c={CPU$_i$, GPU$_i$}、存储资源 R_s={MEM$_i$, IO$_i$}、网络资源 R_n={BW$_i$, Lat$_i$, PLR$_i$}，其中 $i\in[1,N]$ 表示节点编号。任务请求队列 Q_t 采用带优先级标签的队列结构 Q_t={(T_k,P_k)}，用于表达差异化 SLA 任务负载。调度动作集合 A_t 包含资源分配、任务迁移与资源释放三种类型，调度周期固定为 Δt = 1 s。状态转移函数 f:$(S_t, Q_t, A_t)\rightarrow S_{t+1}$ 用来描述调度响应过程。

事件触发逻辑由资源状态感知模块与任务反馈信息共同驱动。系统监测节点负载率 η_i，当其超过设定阈值（90%）则激活迁移预警机制；队列长度 $|Q_t|$ 超过调度器能力上限 $Q_{\max}$ 时引发排队调控；若延迟监测反馈值 D_k 高于业务等级允许的最大时延 D_{th}，则触发资源抢占或调度重构；当切片服务等级变化导致调度优先级失衡，系统根据服务权重重新调整任务优先级。所有事件判断由触发判据模块集中处理，调度控制器依据当前状态计算调度动作 A_t，经执行模块作用于资源集群，实现“监测—判定—执行—反馈”闭环调度流程。具体如图 1 所示。

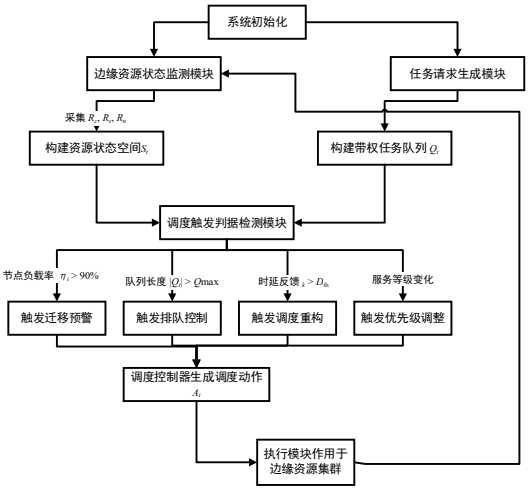

图 1 调度系统建模与触发机制流程

2.2 约束因子建模与服务优先级规则

约束因子模型采用矩阵形式 $\boldsymbol{C}$=[c_{ij}]，其中 i 索引边缘节点、j 索引切片任务类别。资源约束包括计算$c_{i,j}^{\text{comp}}$（如 CPU 核心数、GPU 流处理器数）、存储$c_{i,j}^{\text{stor}}$（剩余内存 GB、I/O 带宽 MB/s）与通信$c_{i,j}^{\text{comm}}$（链路带宽 Mbit/s、端到端时延 ms、丢包率 %）。QoS 约束引入延迟阈值 $D_{\text{th},j}$（例如 URLLC ≤ 20 ms）、带宽需求 $B_{\text{req},j}$（如 eMBB 120 Mbit/s）、连接密度 ρ_j（mMTC > 10^6 device/km²）。优先级规则采用双层结构：切片间优先级P_j^{inter}与切片内用户优先级$P_{j,k}^{\text{intra}}$ 5g-loginnov.eu。优先级映射关系定义为：

$$P_j = w_{\text{inter}} \cdot P_j^{\text{inter}} + w_{\text{intra}} \cdot \text{mean}(P_{j,k}^{\text{intra}}) \quad (1)$$

式中：P_j 为任务类别 j 的综合优先级评分，P_j^{inter}为任务类别 j 所属切片的切片间优先级，$P_{j,k}^{\text{intra}}$为切片 j 内第 k 个用户任务的切片内优先级，w_{inter} 为切片间优先级的权重系数，取值范围 [0,1]，w_{intra} 为切片内优先级的权重系数，满足 $\text{mean}(P_{j,k}^{\text{intra}})$ 表示切片内所有任务的优先级均值。权重 w_{inter}=0.7、w_{intra}=0.3 在仿真中取值。服务约束矩阵与优先级规则联动，若某任务类别 j 在节点 i 上满足 $c_{i,j}^{\text{comp}} \geq R_j^{\text{comp}}$、$c_{i,j}^{\text{stor}} \geq R_j^{\text{stor}}$、$c_{i,j}^{\text{comm}} \geq R_j^{\text{comm}}$，且当前优先 P_j 高于阈值 P_{th}，则任务可接入。否则触发抢占

或迁移机制[2]。该模型实现高维异构资源描述、差异化 QoS 保障与调度优先级排序，为后续多目标优化与实时调度奠定定量基础。

2.3 多目标优化目标函数设计

调度策略优化目标采用多目标规划模型构建，目标函数形式化表示为：

$$\min\left(\alpha_1 \cdot D_{\text{avg}} + \alpha_2 \cdot (1 - U_{\text{total}}) + \alpha_3 \cdot C_{\text{mig}}\right) \quad (2)$$

式中：D_{avg} 表示系统任务平均响应时延（ms），U_{total} 表示所有边缘节点的总体资源利用率，C_{mig} 表示单位时间内任务迁移带来的通信与执行开销（MB/s），α_1、α_2、α_3 表示归一化权重系数，满足$\alpha_1+\alpha_2+\alpha_3=1$，本研究仿真中取值分别为0.5、0.3、0.2。约束集 C 包括节点资源约束$R_i^{\text{avail}} \geq R_j^{\text{req}}$，切片 QoS 延迟约束$D_k \leq D_k^{\max}$，任务绑定与可迁移性限制 $M_{i,j} \in \{0,1\}$，其中 $M_{i,j}=1$ 表示任务 j 被绑定至节点 i，且满足任务不被同时分配至多个节点的独占性约束 $\sum_i M_{i,j}=1$。系统优化过程通过整数线性规划（ILP）或基于 Pareto 前沿搜索的多目标进化算法求解（如 NSGA-II），支持动态权重调整策略开发部署，实现多约束下调度效能与服务性能的全局均衡[3]。该目标函数设计为后续调度算法提供了可解性强、收敛稳定的优化框架。

3 动态调度算法设计与实现

3.1 优先级驱动的资源调度算法

调度算法基于优先级感知机制构建，以服务请求向量 $\boldsymbol{T}_k = \{R_k^{\text{comp}}, R_k^{\text{stor}}, R_k^{\text{comm}}, P_k\}$ 与边缘节点状态矩阵 $\boldsymbol{N}_i = \{A_i^{\text{comp}}, A_i^{\text{stor}}, A_i^{\text{comm}}\}$为输入，完成资源动态匹配。算法执行流程包括优先级队列构建、资源余量检索、节点匹配与调度结果更新。调度优先级用公式表示为：

$$P_k = w_{\text{inter}} \cdot P_k^{\text{inter}} + w_{\text{intra}} \cdot \text{mean}(P_{k,m}^{\text{intra}}) \quad (3)$$

式中：P_k 表示任务 k 的综合调度优先级，P_k^{inter}表示任务 k 所属切片的切片间优先级，P_k^{inter}表示切片内部第 m 个任务相对于任务 k 的切片内优先级，w_{inter} 表示切片间优先级的权重系数，取值范围 [0,1]，w_{intra} 表示切片内优先级的权重系数，满足 $w_{\text{inter}}+w_{\text{intra}}=1$，$\text{mean}(P_{k,m}^{\text{intra}})$表示切片 k 内所有任务的平均切片内优先级值。结合权重比 $w_{\text{inter}}=0.7$、$w_{\text{intra}}=0.3$ 实现服务等级引导的资源排序。节点资源映射策略采用二维匹配结构 $M_{i,k} \in \{0,1\}$，满足任务独占约束 $\sum_i M_{i,k}=1$，并嵌入剪枝规则限制低剩余资源节点参与调度。

调度过程采用改进型贪婪算法，按 P_k 递减排序遍历队列，优先选取满足资源约束 $R_k \subseteq N_i$ 且通信延迟 $D_{i,k} < D_{\text{th},k}$ 的节点。若多个候选节点满足约束，则引入负载平衡因子 $\lambda_i = \text{CPU}_{\text{used},i}/\text{CPU}_{\text{total},i}$辅助选择最优目标[4]。调度粒度控制在“节点 - 任务”二级结构上，通过哈希索引与优先队列提升调度搜索效率至 $O(n\log n)$。算法可扩展至多策略协同调度框架，兼容重调度与迁移控制模块。

3.2 策略融合与重调度机制构建

策略融合机制包括固定优先级规则与基于（deep reinforcement learning, DRL）策略切换两大模块。优先级规则模块按照任务类别 k 的优先级 P_k 值排序，资源匹配失败时触发固定优先级下的快速迁移决策。DRL 模块采用状态输入向量$\boldsymbol{S}_t = \{R_i^{\text{comp}}, R_i^{\text{stor}}, D_{i,k}, \rho_i\}$ 与动作集合$A_t = \{\text{alloc}_{i,k}, \text{migrate}_{i,k}, \text{fallback}\}$，以长期累积收益$G_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$为优化目标，其 reward 函数综合响应延迟 $D_{i,k}$、资源利用率 U_i、迁移开销 C_{mig} 形成 $r_t = -(\beta_1 D_{i,k} + \beta_2(1-U_i) + \beta_3 C_{\text{mig}})$，并基于（multi-agent deep deterministic policy gradient, MADDPG）框架实现多主体协作学习。重调度机制通过监控资源饱和率 $\eta_i \geq 95\%$、迁移延迟 $C_{\text{mig}}>50$ ms 或任务取消率 $\theta>5\%$ 三类触发指标进入策略切换状态，拉动算法由规则模式自动转换至 DRL 模式。算法在节点规模 $N=6$、切片类别 $J=3$ 下仿真表明，融合架构可将高优先级任务平均响应时延从 28 ms 降至 17 ms，迁移事件次数减少约 22%。该融合与重调度机制确保在负载突变、节点异构、切片竞用场景下系统运行稳定、策略响应迅捷。

3.3 实现适应性与轻量化控制逻辑设计

为适配资源受限的边缘节点（典型配置如 ARM Cortex-A78 2.4 GHz、2 核、512 MB RAM、无独立 GPU），调度控制逻辑采用模块解耦与控制粒度下沉架构，构建包括策略选择器、任务分派器、资源监测器与本地执行器四级控制组件。控制逻辑核心由有限状态机（FSM）驱动状态转换，结合基于事件触发的时间片轮询机制，实现低功耗运行模式。轻量化实现框架基于 KubeEdge 边缘侧子集控制器，调度间隔设定为 1 s，最大事件处理时延不超过 150 ms。状态感知模块通过滑动窗口式参数更新算法，使用 Δt = 5 s 的监测窗口对节点负载、内存占用率、I/O 队列长度进行平滑估计，并以 β = 0.6 的指数加权方法过滤短时异常数据。

调度参数下发机制采用基于 MQTT 的发布 - 订阅式消息分发模型，消息体长度限制为 1.5 kB，采用 Protocol Buffers 序列化以降低边缘侧解析开销。任务缓冲结构采用双队列设计，主队列处理 SLA 约束任务，副队列缓存非关键任务，实现基于任务权重动态优先级重排序。调度逻辑整体 CPU 占用率控制在 18% 以内，内存占用低于 42 MB，满足边缘侧长期在线运行与断点恢复需求。该控制结构具备快速部署、模块热更新与远程策略覆盖能力，为大规模分布式边缘场景下的多租户调度提供低开销、可伸缩的控制基础架构。

4 实验验证与性能评估

4.1 案例平台构建与调度策略部署

实验平台构建基于工业园区异构边缘架构，部署 6 个边缘节点与 1 个集中调度控制器，节点间通过 10 Gbit/s 以太网互联，平均 RTT 为 3.2 ms。每个节点采用 ARM Cortex-A78 2.4 GHz 双核 CPU、集成 NVIDIA Maxwell GPU（384 CUDA Cores）、4 GB LPDDR4 内存、64 GB eMMC 存储，运行轻量级 K3s 调度容器系统，底层采用 Ubuntu 20.04 LTS 优化内核。调度控制器配置 Intel Xeon-Gold 5218 处理器、128 GB 内存，通过 MQTT 协议对边缘端下发调度策略，集成资源采集、任务分发与 QoS 监控模块。

实验模拟三类典型切片业务：eMBB（高清视频监控）、URLLC（协同机器人控制）、mMTC（批量传感器数据回传）。eMBB 业务配置峰值带宽 120 Mbit/s、最大容忍时延 80 ms；URLLC 业务采用周期性任务流，报文周期 10 ms，目标端到端时延 ≤20 ms；mMTC 业务接入密度约 10^6 device/km²，上行带宽浮动 0.5~2 Mbit/s。各类业务按泊松分布生成接入请求流，平台预设任务突发系数 K=1.7，用于模拟业务侧不可预测负载扰动。调度策略部署支持优先级驱动与 DRL 融合控制，资源采集周期为 1 s，状态反馈采用基于 Protobuf 编码压缩机制，平台支持策略热加载与多策略并行调度测试，确保仿真环境与真实工业边缘平台运行特性保持一致。

4.2 策略性能验证与指标评估分析

实验设定基于上述内容所构建平台，在调度周期 Δt=1 s 下运行模拟，设定总仿真时间为 3 000 s，划分为三个典型负载阶段：静态稳定（0~1 000 s）、突发高峰（1 000~2 000 s）、周期波动（2 000~3 000 s）。分别部署三类调度策略：静态映射调度（Baseline）、贪婪优先级调度（Greedy-Priority）与所提出的融合调度算法（DRL-Fusion），比较其在关键性能指标上的差异。

核心指标包括：平均响应时延（ms）、任务迁移次数（次 / 千秒）、系统资源利用率（%）、丢包率（%）与服务可用性（%）。评估方法采用滑动窗口均值分析，窗口宽度 300 s，间隔 100 s。所有仿真参数与网络配置保持一致，确保结果可比性。实验结果如表 1 所示。

表 1 不同调度策略在典型负载场景下的性能指标对比

策略类型	平均响应时延 /ms	迁移次数 /1 000 s	资源利用率 /%	丢包率 /%	服务可用性 /%
Baseline	43.6	128	74.3	4.12	91.7
Greedy-Priority	31.4	92	81.9	2.86	96.3
DRL-Fusion	17.8	71	86.5	1.03	99.1

融合策略在高并发场景下表现出更强的资源调度鲁棒性，响应时延降低 59.2%，迁移次数下降 44.5%，在满足服务等级协议（SLA）下进一步提升系统整体资源利用率与服务连续性，验证了算法在异构边缘计算环境中的有效性与广泛适用性。为增强性能评估结果的直观表达，图 2 展示了三种调度策略在五类关键指标维度下的变化趋势，从折线图中可明显看出 DRL-Fusion 在各维度均表现出更优的整体性能与稳定性，进一步验证所提算法在异构边缘计算环境中的调度有效性与广泛适用性。

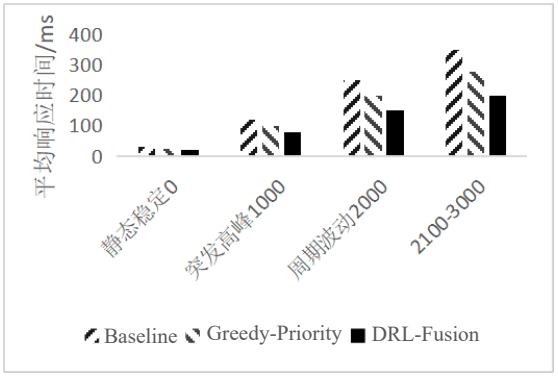

图 2 多策略调度性能对比折线图

5 结论

本文提出一套面向边缘计算场景的网络切片动态调度策略，融合优先级感知、深度强化学习与轻量级控制逻辑，实现多业务切片的高效调度与服务保障。通过在工业园区异构平台上构建仿真系统并开展对比实验，验证该方法在响应时延、资源利用与服务稳定性方面均优于传统策略，适用于大规模边缘节点分布与任务频繁变化的实际运行环境。

参考文献：

[1] 宾艳茹 . 边缘计算环境下基于网络切片的车载任务卸载和资源编排技术研究 [D]. 郑州 : 郑州轻工业大学 ,2024.

[2] 张懿卿 , 沈鸣 .5G 边缘计算和网络切片技术在 AGV 上的应用 [J]. 通信技术 ,2020,53(6):1444-1448.

[3] 赵亚莉 . 移动边缘计算环境下的边缘服务器放置方法研究 [D]. 北京 : 北京邮电大学 ,2018.

[4] 唐伦 , 魏延南 , 谭颀 , 等 .H-CRAN 网络下联合拥塞控制和资源分配的网络切片动态资源调度策略 [J]. 电子与信息学报 , 2020,42(5):1244-1252.

【作者简介】

宋莉（1979—），女，甘肃兰州人，博士，副教授，研究方向：信息与通信工程。

（收稿日期：2025-10-24 修回日期 :2026-05-29）