

基于 IPv6+ 的智能车联算网调度方案设计与实现

于朝阳¹, 辛亮^{1,2}, 杨洋³, 周光涛¹, 马楠⁴

(1. 联通智网科技股份有限公司车辆智能网联研究院, 北京 101599; 2. 北京邮电大学计算机学院, 北京 100876;
3. 中国联合网络通信有限公司研究院, 北京 100048; 4. 北京邮电大学信息与通信工程学院, 北京 100876)

摘要: 针对车联网中算力基础设施成本高、IPv4 网络可编程性差、资源调度灵活性不足的问题, 提出了基于 IPv6+ 的智能车联算网调度方案。通过算网接入、算网感知和算网调度的模块化设计, 实现对“端-边-云”算力、网络及应用的精准感知与高效调度。算网调度模块基于原子编排构建场景化调度策略, 利用强化学习 A2C 算法动态优化任务卸载位置与算力主频, 权衡能耗与时延。同时, 结合 SRv6 可编程路由技术自主选择最优传输路径, 实现基于 IPv6+ 的算网协同优化。仿真实验表明了 A2C 算法支持系统多目标优化的有效性和优越性, 实地部署验证了所提方案在资源灵活调度和降低算力基础设施成本方面的可行性与实用性, 为车路云协同体系建设提供了兼具理论创新与工程价值的解决方案。

关键词: 车联网; IPv6+; 算网调度; 强化学习

中图分类号: U463.67; TP393.02

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2025155

Design and implementation of an IPv6+ based intelligent computing-network scheduling scheme for Internet of vehicles

YU Zhaoyang¹, XIN Liang^{1,2}, YANG Yang³, ZHOU Guangtao¹, MA Nan⁴

1. Intelligent and Connected Vehicle Research Institute, China Unicom Smart Connection Technology Limited, Beijing 101599, China
2. School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China
3. China Unicom Research Institute, Beijing 100048, China
4. School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract: To address the challenges of high infrastructure costs, poor IPv4 network programmability, and insufficient resource scheduling flexibility in Internet of vehicles, an IPv6+ based intelligent computing-network scheduling scheme was proposed. A modular design encompassing computing-network access, awareness, and scheduling was employed to achieve precise perception and efficient coordination of “end-edge-cloud” computing, network and application. The scheduling module constructed scenario-specific strategies based on atomic orchestration, leveraging the A2C reinforcement learning algorithm to dynamically optimize task offloading locations and computing frequencies, thereby balancing energy consumption and latency. Furthermore, it integrated SRv6 programmable routing technology to autonomously select optimal transmission paths, realizing IPv6+ based computing-network synergistic optimization. Simulation results demonstrate the effectiveness and superiority of the A2C algorithm in supporting multi-objective optimization. Field deployments validate the feasibility and practicality of the proposed scheme in enabling flexible resource scheduling and reducing computing infrastructure costs, while providing a solution with both theoretical innovation and engineering value for constructing collaborative vehicle-road-cloud systems.

Keywords: IoV, IPv6+, computing-network scheduling, reinforcement learning

收稿日期: 2025-06-06; 修回日期: 2025-08-18

通信作者: 周光涛, zhouguangtao@chinaunicom.cn

基金项目: 国家重点研发计划基金资助项目 (No.2023YFB4301900, No.2022YFB4300400)

Foundation Items: The National Key Research and Development Program of China (No.2023YFB4301900, No.2022YFB4300400)

0 引言

车联网 (IoV, Internet of vehicles) 作为智能交通系统的核心支撑技术, 为海量实时数据的低时延处理和智能决策提供了重要基础。但是, 车联网场景下的业务需求具有高动态、低时延和高可靠特征, 其中某些场景如远程驾驶对于算力和网络的时延、可靠性和快速调优提出了更高的要求。近年来的研究逐渐关注算力与网络的深度融合, 构建了算力感知网络架构^[1]。然而, 当前网络架构面临多重挑战。其一, 基础设施建设成本高昂, 且分布式算力资源利用率不足。大量路侧多接入边缘计算 (RS-MEC, roadside multi-access edge computing) 单元的部署成本很高且算力有限, 边缘云算力充足但利用率较低。其二, IPv4 网络的可编程性与灵活性受限, 难以支持动态业务需求。面对车联网高并发任务数据传输和低时延服务迁移场景, 需要根据应用需求与网络拓扑等实时调整转发策略。然而, IPv4 的静态路由配置在一定程度上导致了网络功能僵化, 无法实现动态路由优化。其三, 现有任务卸载和资源调度机制缺乏对异构资源 (算力、网络和应用) 的全局协同能力, 难以平衡能耗与时延等多目标优化。

目前已有多项工作围绕上述挑战进行了研究。文献[2]全面阐述了算网调度中的资源管理、跨域调度等基础理论和国内外研究现状。文献[3]通过算网的编排调度解决资源管理和调度问题, 提高了算力节点性能和资源利用率。文献[4]针对算力网络场景需求和融合调度机制进行了探讨。文献[5]从异构资源和多目标优化出发, 讨论了算网融合调度问题。中国联通提出了服务于算网协同和算网一体化需求的新型网络架构^[6]。文献[7]探究了基于视联网人工智能 (AI, artificial intelligence) 业务的算网调度方案。然而, 车联网下的算网融合编排调度需满足高精度感知、低时延通信与高效算网协同调度等核心需求, 目前还缺乏系统性研究。

面对 IPv4 网络的局限性, 一些工作致力于探究基于 IPv6 数据平面的分段路由 (SRv6, segment routing IPv6) 技术的应用。文献[8]提出了面向离散部署的部分 SRv6 网络流量工程算法, 最小化网络最大链路利用率。文献[9]开发了 SRv6 编程模型, 用于保证物联网流量的可靠性和服务质量 (QoS, quality of service)。文献[10]利用 SRv6 网络

可编程特性, 实现无损工作负载迁移, 保障边缘节点之间服务的连续性。以上工作为车联网场景下任务卸载和资源调度过程中的数据转发提供了基础。

车联网场景下的资源调度机制研究已有较为丰富的技术积累, 如结合粒子群优化和模拟退火的混合算法^[11]、基于优先级的任务调度算法^[12]、增强型非支配排序遗传算法^[13]、启发式算法^[14]等。但是这些传统算法通常依赖静态环境或离线优化模型, 难以适应车联网场景下动态变化的复杂环境。

随着人工智能在大规模数据处理、分类和智能决策中的卓越性能表现, 机器学习、强化学习 (RL, reinforcement learning) 等智能算法被引入解决车联网资源调度问题。文献[15]提出了自动驾驶场景下多任务深度强化学习的并行任务调度方法。也有学者利用分层 RL^[16]降低时延、基于深度 Q 网络 (DQN, deep Q-network) 算法最小化任务完成时间^[17]、通过多智能体深度确定性策略梯度算法^[18]和基于 RL 优化分布式计算卸载^[19]减少任务处理时延和能耗, 采用多智能体软演员-评论家算法^[20]进行频谱共享和功率控制。RL 算法通过与环境动态交互, 以端到端的方式自主优化多目标决策, 表现出较好的鲁棒性与适用性。

同时, 人工智能大模型和 6G 技术的突破性发展也为资源智能调度提供了全新范式。文献[21]提出了基于大模型与改进人工蜂鸟算法的网络部署方案, 节省网络资源。文献[22]分析了基于大模型算法在云资源调度领域的优势。然而, 以上算法多侧重单一类型资源 (如算力或网络), 而不是对算力和网络协同优化。文献[23]和文献[24]分别探讨了面向 6G 的算网融合关键技术以及跨域资源编排和分配机制, 但均未深入车联网这一高动态、强实时性的特定场景。此外, 现有面向大模型与 6G 技术的研究大多处于仿真验证阶段, 尚未拓展至实际部署与系统验证。

针对上述挑战, 本文立足于车联网场景构建了一个兼具高灵活性、可编程性、自适应性且可实际部署的算网调度方案, 并为下一代互联网中的算网一体化高效资源管理与调度提供技术参考。具体来讲, 本文主要的研究工作如下。

1) 在架构设计层面, 建立模块化智能车联网算

网调度方案,实现端边云协同。通过算网接入、算网感知、算网调度与能力开放模块的协同工作,实现端边云网资源的精准感知和分布式算力网络下任务的灵活调度,形成跨“端-边-云”的多级异构算力智能调度。

2) 算网调度层面,构建基于原子编排能力的场景化调度策略和基于RL的智能调度算法。将复杂的业务逻辑拆分,通过应用程序接口(API, application program interface)调用和组合,面向不同的业务场景形成调度策略模板按需调用。通过强化学习优势演员-评论家(A2C, advantage actor-critic)算法动态调整任务卸载位置与算力主频参数控制,实现任务卸载中系统能耗和时延多目标优化。

3) 在网络传输层面,采用SRv6技术实现基于IPv6+的可编程路径转发。根据调度模块的决策结果,通过路由实现请求与传输路径的匹配转发,结合网络实时状态与QoS等多个因素,完成高效任务分发与结果回传。

4) 在算法仿真与系统部署层面,仿真实验验证了A2C算法在多目标优化中的有效性,能在线学习并适应复杂动态环境,开展任务调度决策。实地系统部署和性能测试展现了本文方案在资源精细化管理、算力灵活调度以及降低算力基础设施建设成本方面的广阔前景。

1 系统架构

1.1 分层式系统架构

针对算力基础设施建设成本高、IPv4网络架构可编程性差、分布式算力资源利用率不足的问题,本文提出了基于IPv6+的智能车联算网调度方案。如图1所示,此分层式系统架构采用南北协同和多层联动的模块化设计,南向对接基础设施,连接摄像头等感知基础设施、SRv6网关等网络基础设施及RS-MEC等算力基础设施。北向通过标准化接口实现无缝集成,支撑车联网应用,构建“端-边-云”智能调度体系。

系统的4层架构设计如下。

1) 算网接入。构建异构资源纳管平面,实现端边云等基础设施的接入,负责感知基础设施、中心云、边缘云、RS-MEC等多级算力和SRv6网关等网络设备的接入,实现端边云资源的采集、整合与控制。

2) 算网感知。建立基于IPv6+的全域资源状态感知体系,实现算力感知、网络感知和应用感知多维度监控,为算网调度提供数据支持。算力感知包括对中心云、边缘云与RS-MEC的通用算力-中央处理器(CPU, central processing unit)、智算算力-图形处理器(GPU, graphics processing unit)和神

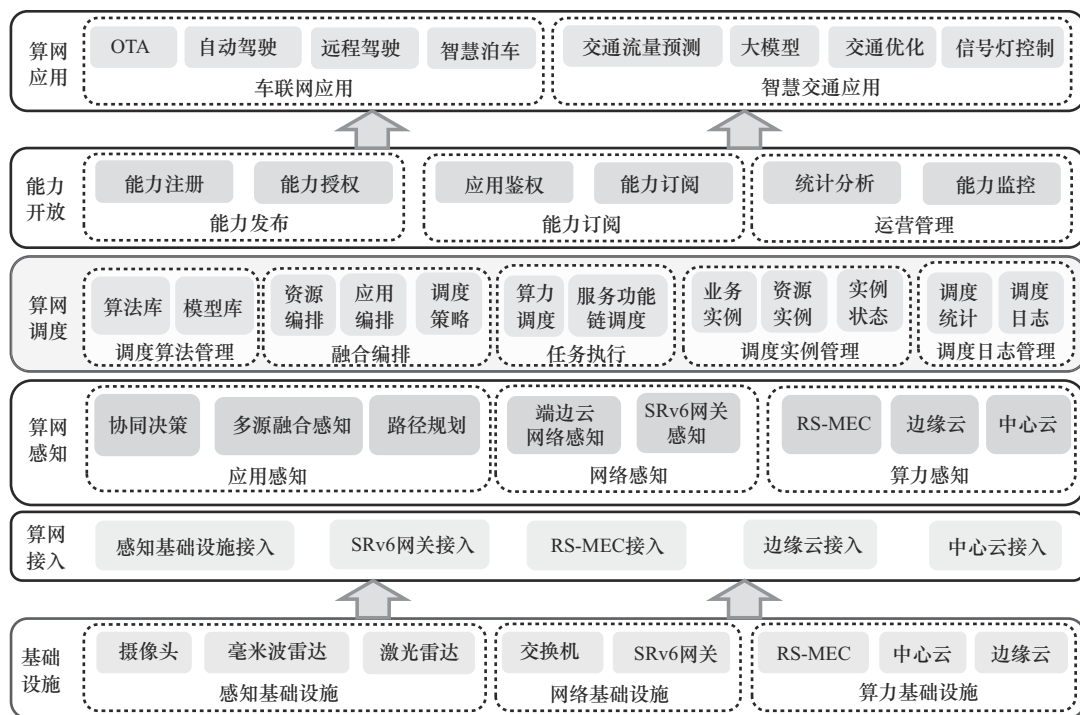


图1 分层式系统架构

经网络处理器 (NPU, neural processing unit)、内存 (Mem, memory) 等计算和存储资源的实时资源利用率进行感知; 网络感知包括对 SRv6 网关设备及端边云各级之间的网络状态 (如时延、抖动和丢包率等) 进行感知; 应用感知对部署在端边云等算力设备上的车联网应用 (如多源融合感知、协同决策、路径规划等服务状态) 进行感知。

3) 算网调度。基于原子编排能力构建场景化调度策略, 通过集成调度算法管理、融合编排、任务执行、调度实例管理和调度日志管理模块, 实现智能调度决策, 优化资源配置。调度算法管理实现模型库 (模型镜像、应用镜像等) 和算法库 (函数算法) 的管理; 融合编排提供原子维度的编排能力, 实现对资源编排、应用编排和调度策略的管理, 完成面向端边云业务场景的编排设计; 任务执行通过 SRv6 技术进行智能选路与转发, 面向端边云场景的调度, 分为算力调度和服务功能链 (SFC, service function chain) 调度; 调度实例管理实现对业务实例、资源实例和实例状态的管理; 调度日志管理记录调度的执行情况, 便于运维和管理。

4) 能力开放。聚焦能力发布、能力订阅与运营管理, 将车联网算网调度系统的算网调度和业务调度能力进行封装, 促进能力共享与利用。能力发布实现将算网调度感知及业务调度能力封装; 能力订阅为面向车联网的算网平台提供能力订阅和应用鉴权; 运营管理实现对算网调度系统的能力监控和统计分析。

基于 IPv6+ 的智能车联网算网调度方案通过分层解耦, 形成算网接入、算网感知、算网调度与能力开放的控制流程, 构建高效、智能、协同的调度系统, 支持自动驾驶、远程驾驶、空中下载 (OTA, over the air) 技术升级、智慧泊车等车联网应用, 以及交通流量预测、大模型、交通优化、信号灯控制等智慧交通应用。

1.2 与现有方案的对比分析

为系统性地展示基于 IPv6+ 的智能车联网算网调度方案的技术优势, 本文选取了若干代表性方案, 从应用场景、关键能力和网络协议 3 个方面进行了对比分析, 如表 1 所示。

由表 1 可以看出, 在应用场景方面, 本文方案为车联网高动态、低时延和高可靠业务需求设计,

区别于通用场景或其他场景, 具有更强的适配性。在关键能力方面, 本文方案同时具备算网调度和网络路径可编程两大核心能力, 能够克服单一资源调度局限, 实现算网协同优化, 并依托 IPv6+ 网络协议, 具备底层网络路径编程能力。因此, 本文方案兼具底层可编程能力和高层调度灵活性, 为车联网复杂场景提供更稳定、高效和智能的算网服务, 具有显著的综合技术优势。

表 1 本文方案与现有方案的对比

参考文献	应用场景	关键能力		网络协议
		算网调度	网络路径可编程	
文献[6]	通用场景	具备	具备	IPv6+
文献[7]	视联 AI 场景	具备	不具备	IPv4
文献[10]	通用场景	不具备	具备	IPv6+
文献[15]	车联网	具备	不具备	IPv4
文献[21]	监测场景	网络调度	不具备	IPv4
本文方案	车联网	具备	具备	IPv6+

2 基于 IPv6+ 的智能车联网算网调度方案

2.1 业务调度流程

针对车联网场景下算力基础设施成本高的问题, 本文提出了智能车联网算网调度方案, 通过端边云资源的协同调度, 降低路侧基础设施成本。业务调度流程共分数据感知、调度触发、调度执行、调度记录与调度恢复 5 个阶段, 如图 2 所示。各阶段的技术实现如下。

数据感知。实时采集端、边、云的算力及网络资源状态与应用服务健康度。1) 探针部署。以端边调度为例, RS-MEC、边缘云以及 SRv6 网关设备均部署感知探针, 用于实时感知设备的资源使用情况 (通用算力、智算算力、内存、存储等)、网络状态 (端到端时延、链路带宽利用率等)、所部署应用服务的健康度 (容器服务心跳间隔等)。2) 数据聚合。算网感知模块聚合探针数据, 生成统一格式的资源状态快照。3) 状态同步。调度模块采用订阅的方式周期性访问算网感知模块, 获取感知数据。

调度触发。基于事件驱动机制, 结合预设的调度策略和算法生成调度决策。1) 条件匹配。感知数据匹配预设策略库中的场景规则触发相应的调度

策略，如RS-MEC的CPU利用率大于80%且网络抖动大于30ms触发跨域卸载。2)策略执行。在相应策略下，选择调度算法（如基于RL的调度算法）。3)决策生成。执行调度算法，得到最终的调度决策（即任务的目的服务端地址），并生成对应的路径集合。

调度执行。通过SRv6网关动态切换数据流路径。根据调度决策，SRv6网关结合实时网络状态和业务需求选择最优路径，并将数据流按照指定路径转发到目的服务端处理。

调度记录。生成调度实例和调度日志，用于流程审计与策略优化，实现调度智能体持续迭代。

调度恢复。在资源恢复正常后切回原始服务节点。1)恢复条件检测。调度模块周期性访问算网感知模块，获取感知数据，设置滞后阈值，防止乒乓效应。满足资源恢复条件后，执行调度恢复操作。2)路径回切。依据SRv6双向路径对称原则，沿源路径反向回传数据流。

兵效应。满足资源恢复条件后，执行调度恢复操作。2)路径回切。依据SRv6双向路径对称原则，沿源路径反向回传数据流。

2.2 算网感知

算网感知模块如图3所示，通过算网接入实现对算力资源、网络资源以及应用服务状态的精细化感知，确保资源的精准管理与高效利用。

面向中心云和边缘云，感知数据包含承载在中心云和边缘云上的集群、节点和容器组（Pod）的资源感知，包括集群、节点的CPU、内存、存储容量、剩余量、分配量、使用率等指标，以及Pod容器CPU、内存、存储总量、分配量、GPU显存分配量等信息。

面向RS-MEC，感知数据包含RS-MEC上节点的感知、承载在RS-MEC上算力服务的感知以及应

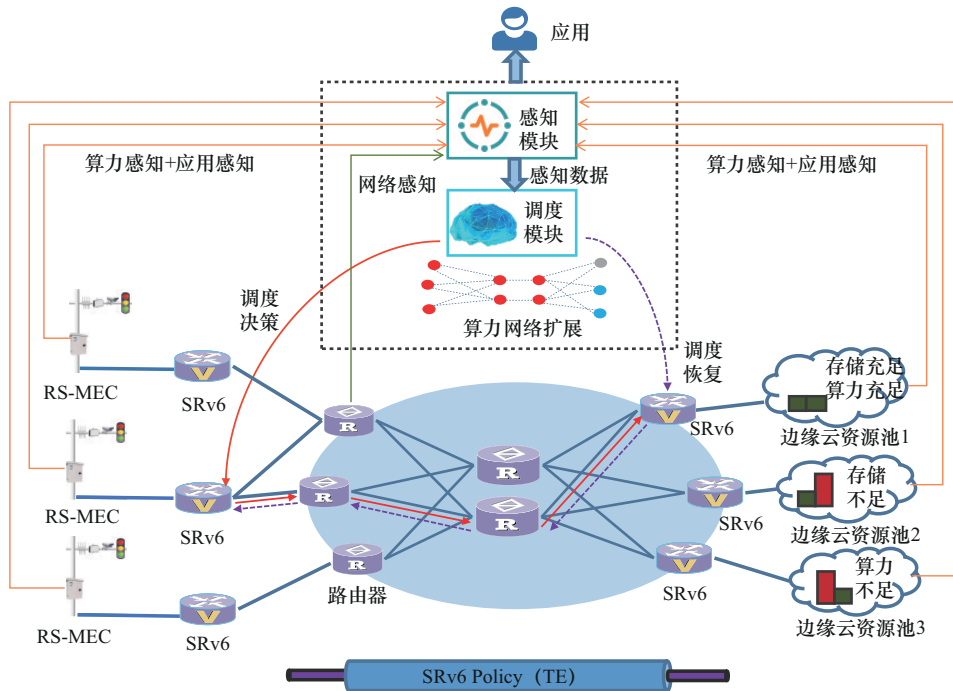


图2 业务调度流程

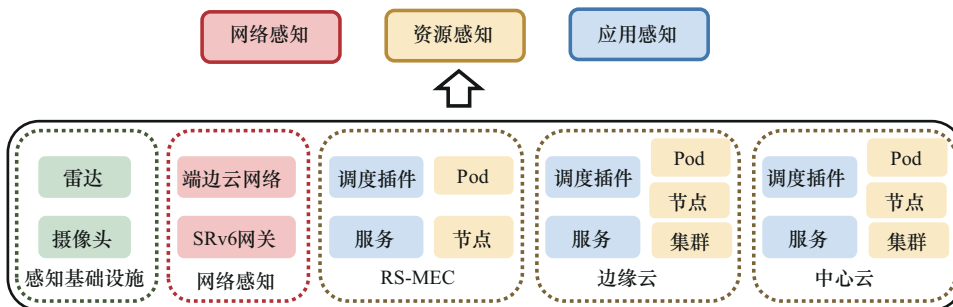


图3 算网感知模块

用服务（如推拉流服务等）的感知。包含 RS-MEC 节点的 CPU、内存、存储、GPU 显存总量、分配量、剩余量、使用率等指标，承载在 RS-MEC 上的 Pod 容器 CPU、内存、存储、GPU 显存分配量、使用率等，以及推拉流服务的状态和算力调度插件的状态。

面向网络资源，网络感知数据包含 SRv6 网关和 RS-MEC-边缘云-中心云的网络质量的感知。其中 SRv6 网关包含 SRv6 网关关联的 RS-MEC 链路信息和网络质量数据（带宽、时延、抖动、丢包等）。RS-MEC-边缘云-中心云的网络感知包含 RS-MEC 到边缘云以及边缘云到中心云的链路信息以及该链路的网络质量数据。

面向感知基础设施，依托推拉流服务，可获取到关联到 RS-MEC 上的摄像头、激光雷达等数据，并可获取到感知设备的数据流，依托数据流判断感知设备的运行状态。

2.3 场景化调度策略

本文构建了基于原子编排能力的场景化调度策略，支持可编程的调度触发机制，通过原子化的 API 编排能力，面向不同的业务场景构建不同的调度策略。调度模块的工作流程如图 4 所示，包含 4 个核心环节。

1) 调度策略。基于原子编排能力构建场景化调度策略，实现多维度资源与服务的动态编排。首先，构建原子化服务资源池，包含智能车联网调度系统提供的 API 或自身的业务封装的 API。其次，基于应用资源指标、算力资源性能指标、网络资源指标等实际需求，加载算法库和模型库（如负载均衡算法、路径优化算法、强化学习算法等），形成场景化的策略模板（如视频流调度、服务扩缩

容、端边数据流调度等）。最后，将策略要素（触发规则、编排逻辑、执行流程等）进行标准化封装，绑定策略后发布。面向车联网不同的业务场景或复杂环境（如不同网络状况、车流量等）可绑定不同的调度策略，保障车联网业务可靠性。例如，针对网络状况良好且车流量较高情况下，设定端边数据流调度策略，配置双阈值触发机制，即当前算力 CPU 利用率达到 80% 且内存利用率达到 40% 时触发端边调度，然后将调度流程中的原子操作（如调用推拉流服务 API、更改应用服务配置、加载调度算法等）编排成策略模板发布。

2) 策略应用。场景编排完成后，系统实时监测多维业务需求与异构资源状态，动态匹配预置策略。当环境参数满足触发条件时，激活对应的策略模板，调用相应的调度算法模型，设计最优调度方案。

3) 任务执行。依托算网接入，调用 SRv6 网关实现智能选路，联动 RS-MEC、边缘云、中心云等资源执行算力调度。执行算力调度过程中，通过资源编排动态分配计算资源。执行服务功能链调度过程中，结合资源编排与应用编排能力，实现服务功能链动态重构。

4) 调度日志。建立全链路追踪机制，完整记录调度全生命周期数据，将触发调度条件、调度时间、调度类型、调度的源节点和目的节点等写入调度日志，生成调度实例与调度日志，记录资源变更详情。

调度模块可支撑的场景有 2 类，分别为集中式调度与分布式调度（包括同层调度与跨层调度）。集中式调度仅集中在单个计算资源如端侧、边缘云或中心云内部，可实现调度应用服务的远程部署、

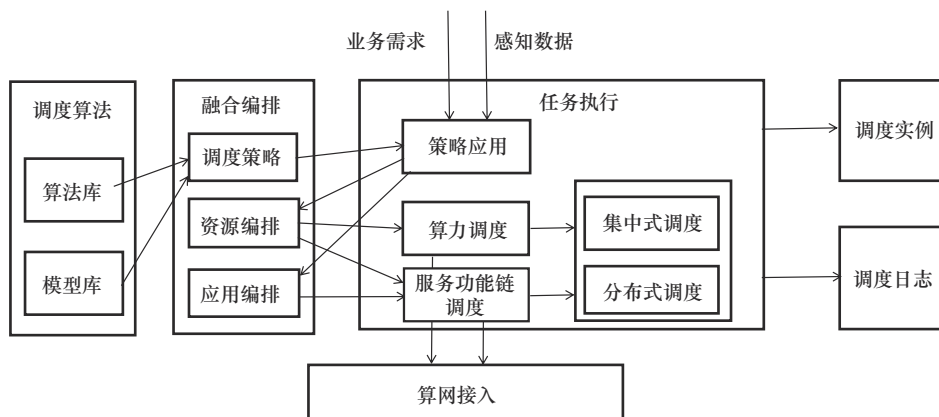


图 4 基于原子编排能力的场景化调度策略

应用服务的扩缩容等。分布式调度涉及多类计算资源，包括端端数据流重定向、端边任务迁移等。

2.4 基于 SRv6 的智能选路与转发机制

为解决 IPv4 网络的可编程性与灵活性受限，难以支持动态业务需求的问题，本文提出的调度方案基于 SRv6 技术进行智能选路与转发，实现网络端到端打通和业务、路径和应用的动态编排。

SRv6 的可编程特性主要包含以下 3 种。

1) 源路由控制。SRv6 改变了传统 IP 网络中基于目的地址的寻路机制，采用源路由编程范式，基于 128 位 IPv6 地址空间定义 SID (segment ID)，支持在报文头部携带显式路径指令，实现路径可编程^[25]。

2) 业务感知路由。将网络服务功能（如 QoS 保障、服务链编排）编码为可编程 SID 序列，实现业务可编程。

3) 状态驱动优化。结合实时网络状态与 QoS 等多个因素调整转发策略。

基于 SRv6 的智能选路与转发机制如图 5 所示，SRv6 网关根据服务端的目的 IP，通过路由实现请求与传输路径的匹配转发，结合实时状态与 QoS 等多个因素选择最优路径，实现高效任务分发与结果回传。

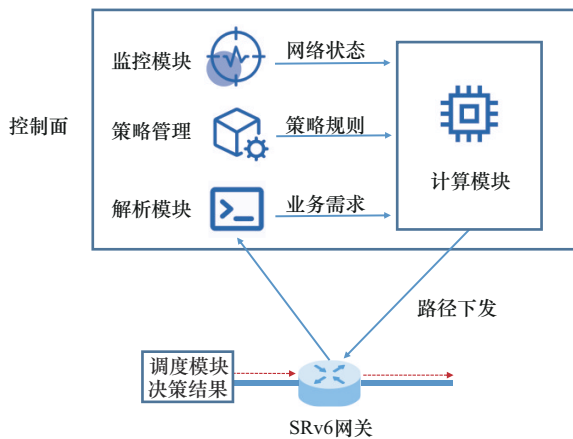


图 5 基于 SRv6 的智能选路与转发机制

调度决策输入。SRv6 网关控制面接受调度模块下发的调度决策，即目的服务端地址。

请求解析。解析模块解析业务需求，包括服务端的 IPv6 地址和端点信息、任务类型和优先级、QoS 约束如时延和带宽等，生成结构化的请求信息。

路径计算。计算模块基于解析模块输出的业务

需求、预定义好的策略规则、监控模块实时采集的网络状态（如链路负载、时延、丢包率等），计算满足 QoS 约束的最优路径。随后，进行 SRv6 路径封装，生成 Segment List，并输出 SRv6 路径方案。

路径反馈。SRv6 网关控制面将 Segment List 下发至数据平面，数据平面更新转发表，绑定服务端目的 IP 与 Segment List 映射关系。

路径建立与任务分发。SRv6 网关封装段路由头 (SRH, segment routing header)，按照 Segment List 将请求逐跳转发至目的服务端的网关，并由目的服务端接受任务并处理。

3 基于 A2C 的任务调度算法

针对现有资源调度机制对异构资源（算力、网络和应用）的全局调度灵活性不足，导致系统能耗与时延难以平衡的问题，本文基于强化学习优势演员-评论家算法进行调度决策，实现任务卸载中系统时延与能耗的多目标联合优化。

3.1 建模

在车联网场景下，RS-MEC 上部署多源融合感知算法等应用服务，基于来自路侧摄像头和激光雷达等多种感知设备的数据进行辅助驾驶决策。以端边调度为例，由于摄像头和激光雷达的数据量大，RS-MEC 的算力有限，面对 RS-MEC 算力不足或者 RS-MEC 出现意外宕机的情况，需要及时将相关的计算任务转移到其他算力充足的 RS-MEC 或边缘云进行处理，保证车联网业务的连续性。同时，需要高效的调度算法，保证整个任务卸载过程中系统能耗和时延的多目标优化。

边缘计算模型包括多个 RS-MEC 和边缘服务器。调度模块结合网络信息、RS-MEC 和边缘服务器算力信息以及应用信息（任务需求），实现计算任务卸载。系统模型设置任务为最小的处理单位，一个任务仅在一个计算节点完成，不能由多个节点协同处理。假定系统共包含 H 个边缘服务器 $S = \{S_1, S_2, \dots, S_H\}$ ，第 h 个边缘服务器可以表示为 $S_h, (1 \leq h \leq H)$ 。同时，系统中共包含 L 个 RS-MEC $M = \{M_1, M_2, \dots, M_L\}$ ，第 l 个 RS-MEC 可以表示为 $M_l, (1 \leq l \leq L)$ ，计算任务 $A \triangleq \{Z, C, \sigma\}$ ，其任务属性包括任务携带的数据量 Z 、处理任务所需的计算量 C 和任务的处理时限 σ 。系统参数如表 2 所示。

表 2 系统参数

参数	含义	参数	含义
M_l	第 l 个 RS-MEC	φ	路径损耗指数
S_h	第 h 个边缘服务器	D_0	参考距离
A	计算任务	$D_{l,h}$	M_l 和 S_h 的距离
Z	任务携带的数据量	$\mu_{l,h}$	M_l 和 S_h 的小尺度衰落信道功率增益
C	处理任务所需的计算量	κ_h	边缘服务器能耗系数
σ	任务的处理时限	κ_l	RS-MEC 能耗系数
Q	任务队列	B	总带宽
η	路径损耗常数	$\varsigma_{l,h}$	RS-MEC M_l 的发射功率
$f_{l,\max}$	RS-MEC 最大功率	$p_{l,h}$	RS-MEC M_l 所占带宽比例
$f_{h,\max}$	边缘服务器最大功率	ρ	噪声功率密度
$G_{l,h}$	M_l 和 S_h 的信道增益	$B_{l,h}$	M_l 和 S_h 的信道容量

边缘服务器和 RS-MEC 中均设置有任务队列 Q ，按照先进先出 (FIFO, first in first out) 的顺序对队列中的任务进行处理。假定边缘服务器 S_h 的任务队列 $Q_h = \{A_{h,1}, A_{h,2}, \dots, A_{h,N_h}\}$ ，其任务数量为 N_h 。假设 S_h 为单核边缘服务器，其在处理队列中第 i 个任务 $A_{h,i}$ 的运行频率为 $f_{h,i}$ ，则第 i 个任务所需的计算时长为

$$T_{h,i}^{\text{comp}} = \frac{C_{h,i}}{f_{h,i}} \quad (1)$$

在 t 时刻，边缘服务器 S_h 正在处理第 j 个任务 $A_{h,j}$ ，此任务剩余数据所需的计算量为 $C'_{h,j}$ 。当前任务队列中待处理的任务数量 $N'_h = N_h - j + 1$ 。此时，将新的任务卸载到边缘服务器 S_h ，记为 A_{h,N_h+1} ，其等待时长为

$$T_{h,N_h+1}^{\text{wait}} = \frac{C'_{h,j}}{f_{h,j}} + \sum_{i=j+1}^{N_h} \frac{C_{h,i}}{f_{h,i}} \quad (2)$$

任务 A_{h,N_h+1} 卸载到 S_h 后的周转时长包含等待时长和计算时长，可以表示为

$$T_{h,N_h+1}^{\text{turn}} = T_{h,N_h+1}^{\text{wait}} + T_{h,N_h+1}^{\text{comp}} \quad (3)$$

服务器功耗可近似与 CPU 运行频率的三次方成正比，设边缘服务器能耗系数为 κ_h ，为了表达简洁，在时间 t 内边缘服务器 S_h 的能耗表示为

$$E_h^{\text{comp}} = \sum_{i=1}^{j-1} \kappa_h f_{h,i}^3 T_{h,i}^{\text{comp}} + \kappa_h f_{h,j}^3 \frac{C_{h,j} - C'_{h,j}}{f_{h,i}} = \sum_{i=1}^{j-1} \kappa_h f_{h,i}^2 C_{h,i} + \kappa_h f_{h,j}^2 (C_{h,j} - C'_{h,j}) \quad (4)$$

同理，当任务卸载到第 l 个 RS-MEC M_l 时，将任务记为 A_{l,N_l+1} ，其卸载到 M_l 的总处理时长为

$$T_{l,N_l+1}^{\text{turn}} = T_{l,N_l+1}^{\text{wait}} + T_{l,N_l+1}^{\text{comp}} \quad (5)$$

RS-MEC 的能耗为

$$E_l^{\text{comp}} = \sum_{i=1}^{j-1} \kappa_l f_{l,i}^2 C_{l,i} + \kappa_l f_{l,j}^2 (C_{h,j} - C'_{h,j}) \quad (6)$$

设定 RS-MEC M_l 到边缘服务器 S_h 的小尺度衰落信道功率增益为 $\mu_{l,h}$ ，满足 $E[\mu_{l,h}] < \infty$ 。那么 M_l 和 S_h 之间的信道增益可以表示为 $G_{l,h} = \mu_{l,h} \eta \left(\frac{D_0}{D_{l,h}}\right)^\varphi$ ，其中 η 和 φ 分别表示路径损耗常数和指数， D_0 和 $D_{l,h}$ 分别表示参考距离以及 M_l 和 S_h 的距离。根据香农定理，RS-MEC 到边缘服务器的信道容量 $B_{l,h}$ 为

$$B_{l,h} = p_{l,h} B \text{lb} \left(1 + \frac{G_{l,h} \varsigma_{l,h}}{p_{l,h} \rho B} \right), p_{l,h} \geq 0, \sum_{l=1}^L p_{l,h} \leq 1 \quad (7)$$

其中， B 表示总带宽， $\varsigma_{l,h}$ 和 $p_{l,h}$ 分别表示 RS-MEC M_l 的发射功率和所占用的带宽比例， ρ 表示噪声功率密度。将任务卸载到边缘服务器时，需要将任务计算相关的数据从 RS-MEC 传输到边缘服务器，数据从 RS-MEC M_l 传输到边缘服务器 S_h 所需的时长为

$$T_{h,N_h+1}^{\text{trans}} = \frac{Z_{h,N_h+1}}{B_{l,h}} \quad (8)$$

任务 A_{h,N_h+1} 由 RS-MEC M_l 传输到边缘服务器 S_h 的传输能耗为

$$E_{h,N_h+1}^{\text{trans}} = \varsigma_{l,h} T_{h,N_h+1}^{\text{trans}} \quad (9)$$

3.2 算法设计

本文采用 A2C 算法构建决策优化智能体，进行车联网场景下的智能调度决策。A2C 算法作为强化学习中一种常用的优化算法，结合了策略梯度方法和价值函数逼近的优点，通过引入优势函数改进传统的演员-评论家算法。A2C 算法智能体中包含 2 个网络。

1) 演员 (Actor)。即策略网络，用 $\pi(a|s; \theta)$ 表示，其中 θ 表示策略网络的参数，Actor 负责选择动作，根据策略在状态 s 下，选择最优的执行动作 a 。

2) 评论家 (Critic)。即价值网络，用 $v(s; \omega)$ 表示，其中 ω 表示价值网络的参数，Critic 负责评

估 Actor 当前策略的好坏。

A2C 算法流程如图 6 所示。感知模块的感知信息作为调度模块的智能体输入，感知状态包含应用信息（来自 RS-MEC 的任务信息）、算力信息（RS-MEC 和边缘服务器的负载状态）以及网络信息（RS-MEC 和边缘服务器的带宽）。智能体根据感知状态做出决策动作并执行，环境对智能体的动作给出反馈，并进行质量评估（本文的优化目标为时延和能耗）。智能体依据环境的反馈不断更新策略网络和价值网络。

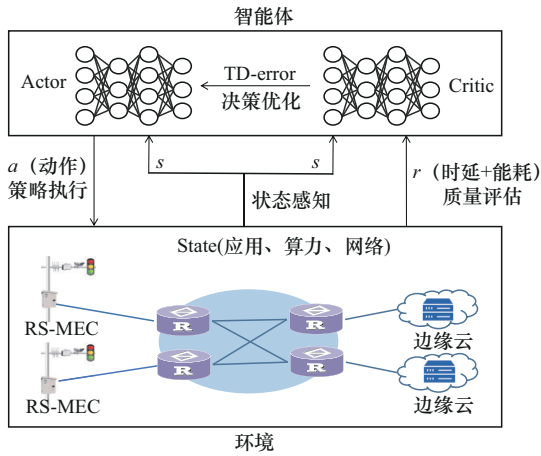


图6 A2C算法流程

在某时刻 t ，系统需要对 RS-MEC M_l 上的任务 A 进行调度决策，智能体感知的环境状态为

$$s_t = \{ A, p_l, \psi, \psi' \} \quad (10)$$

其中， $A \triangleq \{ Z, C, \sigma \}$ 表示当前任务信息（应用状态）， $p_l = \{ p_{l,1}, p_{l,2}, \dots, p_{l,h}, \dots, p_{l,H} \}$ 表示当前任务所在的 RS-MEC 到各边缘服务器的带宽比例（网络状态）， $\psi = \{ N_l, T_l^{\text{wait}}, f_{l,\max}, U_l \}$ ， $1 \leq l \leq L$ 表示当前 RS-MEC 的负载信息（算力状态），包括队列长度 N_l 、

$$\varpi = \frac{\sum_{h=1}^H \left(\sum_{i=1}^{N_h} \zeta_{h,i} + I(S_h) \zeta_{h,N_h+1} \right) + \sum_{l=1}^L \left(\sum_{i=1}^{N_l} \zeta_{l,i} + I(M_l) \zeta_{l,N_l+1} \right)}{\sum_{h=1}^H N_h + \sum_{l=1}^L N_l + 1} \quad (15)$$

3.3 算法流程

智能体基于策略 π 与环境交互，优势函数 $A(s_t, a_t)$ 用于衡量某个动作相对于某一状态下的平均水平的好坏，设为 $A(s_t, a_t) = Q_\pi(s_t, a_t) - V_\pi(s_t)$ ，其中 $Q_\pi(s_t, a_t)$ 是状态-动作值函数，表示在状态 s_t 下执行动作 a_t 后的预期奖励。 $V_\pi(s_t)$ 是状态值函

等待时长 T_l^{wait} 、最大频率 $f_{l,\max}$ 以及当前资源使用率 U_l ，对于边缘服务器 S_h 来说同理，其负载信息可以表示为 $\psi' = \{ N_h, T_h^{\text{wait}}, f_{h,\max}, U_h \}$ ， $1 \leq h \leq H$ 。

智能体为每个任务进行最优动作决策，包括确定任务的卸载位置（RS-MEC 或边缘服务器）及相应的服务器处理此任务的频率，动作为

$$a_t = \{ d_t, f_{l,t}, f_{h,t} \} \quad (11)$$

其中， $d_t = \{ I(M_1), \dots, I(M_L), I(S_1), \dots, I(S_H) \}$ ，表示任务的卸载位置， I 表示任务卸载位置的标志位，满足 $\sum_{h=1}^H I(S_h) + \sum_{l=1}^L I(M_l) = 1$ ，且 $I(M_l) \in \{0, 1\}$ ， $I(S_h) \in \{0, 1\}$ 对于 $\forall h \in \{1, 2, \dots, H\}$ ， $\forall l \in \{1, 2, \dots, L\}$ ，表示任务有且只有一个卸载位置。当 $I(M_l)$ 或 $I(S_h)$ 为 1 时，表示将任务卸载到此 RS-MEC 或边缘服务器上，为 0 时则不卸载。

智能体的优化目标为实现系统能耗和时延的权衡，包括 RS-MEC 和边缘服务器的计算能耗以及 RS-MEC 到边缘服务器的传输能耗，表示为

$$E_t^{\text{total}} = E_{h,N_h+1}^{\text{trans}} I(S_h) + E_h^{\text{comp}} + E_l^{\text{comp}} \quad (12)$$

任务总时延可以表示为

$$T_t^{\text{total}} = (T_{h,N_h+1}^{\text{turn}} + T_{h,N_h+1}^{\text{trans}}) I(S_h) + T_{l,N_l+1}^{\text{turn}} I(M_l) \quad (13)$$

智能体获得奖励（优化目标）为

$$r_t = E_t^{\text{total}} + \lambda T_t^{\text{total}} \quad (14)$$

其中， λ 为权重因子，用于衡量时延在整个优化目标中的重要程度。当 $T_t^{\text{total}} \leq \sigma$ 时，表示此任务（队列中第 i 个任务，卸载位置为 M_l 或 S_h ）按时完成，则令 $\zeta_{l,i} = 1$ 或 $\zeta_{h,i} = 1$ ，否则判定此任务超时，令 $\zeta_{l,i} = 0$ 或 $\zeta_{h,i} = 0$ 。

任务超时率 ϖ 可以表示为

数，表示在状态 s_t 下的预期总奖励。如果 $A(s_t, a_t)$ 为正，说明动作 a_t 优于在状态 s_t 下的平均水平，否则劣于平均水平。训练策略网络的思路是使用策略梯度 $\nabla_\theta J(\theta)$ 更新策略网络参数 θ 。

$$\nabla_\theta J(\theta) = E_S [E_A [\nabla_\theta \ln \pi(a|s; \theta) (Q_\pi(s, a) - V_\pi(s))]] \quad (16)$$

对策略梯度进行蒙特卡罗近似，策略梯度 $\nabla_{\theta} J(\theta)$ 的无偏估计^[26]为

$$g(s, a; \theta) = (Q_{\pi}(s, a) - V_{\pi}(s)) \cdot \nabla_{\theta} \ln \pi(a|s; \theta) \quad (17)$$

根据动作值函数和状态值函数的性质，可得

$$Q_{\pi}(s_t, a_t) = E_{S_{t+1}}[r_t + \gamma \cdot V_{\pi}(s_{t+1})] \quad (18)$$

$$V_{\pi}(s_t) = E_{A_t, S_{t+1}}[r_t + \gamma \cdot V_{\pi}(s_{t+1})] \quad (19)$$

智能体执行动作 a_t 之后，环境反馈新的状态 s_{t+1} 和奖励 r_t ，得到四元组 (s_t, a_t, r_t, s_{t+1}) ，动作价值函数的无偏估计为

$$Q_{\pi}(s_t, a_t) \approx r_t + \gamma \cdot V_{\pi}(s_{t+1}) \quad (20)$$

那么有

$$g(s, a; \theta) = (r_t + \gamma \cdot V_{\pi}(s_{t+1}) - V_{\pi}(s)) \cdot \nabla_{\theta} \ln \pi(a|s; \theta) \quad (21)$$

用价值网络 $v(s; \omega)$ 近似状态值函数 $V_{\pi}(s)$ ，则有 $\tilde{g}(s, a; \theta) = (r_t + \gamma \cdot v(s_{t+1}; \omega) - v(s_t; \omega)) \cdot \nabla_{\theta} \ln \pi(a|s; \theta)$

TD-target 可以表示为

$$\hat{y}_t = r_t + \gamma \cdot v(s_{t+1}; \omega) \quad (23)$$

TD-error 可以表示为

$$\delta = v(s_t; \omega) - \hat{y}_t \quad (24)$$

因此，策略网络的参数 θ 的更新方式为

$$\theta \leftarrow \theta - \beta \cdot \delta \cdot \nabla_{\theta} \ln \pi(a|s; \theta) \quad (25)$$

对于价值网络，Loss 可以表示为

$$L(\omega) \triangleq \frac{1}{2} (v(s_t; \omega) - \hat{y}_t)^2 \quad (26)$$

那么 $\nabla_{\omega} L(\omega) = (v(s_t; \omega) - \hat{y}_t) \nabla_{\omega} v(s_t; \omega)$ ，价值网络的参数 ω 更新方式为

$$\omega \leftarrow \omega - \alpha \cdot \delta \cdot \nabla_{\omega} v(s_t; \omega) \quad (27)$$

基于 A2C 的任务调度算法如算法 1 所示，引入优势函数后，A2C 算法相较于原始的 Actor-Critic 方法可以显著减少策略梯度的方差，从而使策略更新更加稳定。

算法 1 基于 A2C 的任务调度

输入 RS-MEC 队列 M ，边缘服务器队列 S ，奖励折扣 γ ，学习率 α 和 β ，策略网络参数 θ ，价值网络参数 ω ，训练步数 n 和最大训练步数 \mathcal{G}

输出 优化后的策略网络参数 θ ，价值网络参数 ω

初始化 策略网络 π 及其参数 θ ，价值网络 $v(s; \omega)$ 及其参数 ω ，系统状态 $s = s_0$ ，总回报

$R^{\text{total}} = 0$ ，初始训练步数 $n=0$

1) repeat for each step n

2) 获取当前系统状态 s_t

3) 根据当前策略 $\pi(a_t|s_t; \theta)$ 获得动作 a_t

4) 执行动作 a_t ，从环境中获取奖励 r_t 及下一时刻状态 s_{t+1}

5) $R^{\text{total}} += r_t$

6) 计算优势函数（用 TD-error 代替，式(24)）

7) 更新价值网络 $\omega \leftarrow \omega - \alpha \cdot \delta \cdot \nabla_{\omega} v(s_t; \omega)$

8) 更新策略网络 $\theta \leftarrow \theta - \beta \cdot \delta \cdot \nabla_{\theta} \ln \pi(a|s; \theta)$

9) 更新系统状态 $s_t = s_{t+1}$

10) $n = n + 1$

11) until 策略 π 收敛或最大训练步数 $n = \mathcal{G}$

A2C 算法基于实时感知数据（包括算力、网络和应用状态）决策任务卸载位置后，通过 SRv6 编程选择最优传输路径，构建了基于 IPv6+ 的算网协同编排和优化。

4 算法仿真

4.1 仿真环境设置

仿真设备采用 Intel(R) Core(TM) i7-10510U CPU @ 1.80 GHz*8 GPU NVIDIA 556.35，搭建的模拟系统的相关参数取值如表 3 所示，取值参考当前已有的边缘计算相关文献[27]。

表 3 参数取值

参数	取值	参数	取值
$\mu_{l,h}$	exp(1)	$\varsigma_{l,h}/\text{mW}$	500
B/MHz	10	κ_l	10^{-27}
$\rho/(\text{dBm} \cdot \text{Hz})^{-1}$	-174	κ_h	10^{-27}
η/dB	-40	Z/bit	[0, 50 000]
φ	4	C/cycle	[30 000, 50 000]
D_0/m	1	$f_{l,\text{max}}/\text{GHz}$	2.5
$D_{l,h}/\text{m}$	[90, 150]	$f_{l,\text{max}}/\text{GHz}$	1

4.2 仿真性能分析

本文通过调整式(14)中的权重因子 λ 的值来调整系统的优化目标能耗与时延之间的权衡， λ 越大则表示系统侧重对时延的优化， λ 越小则表示系统侧重对能耗的优化。本文选取以下 2 种基线算法验证 A2C 算法的有效性。

1) Rule-based 算法。基于规则的调度算法，每

次均将任务调度到最小等待队列的计算节点中（边缘服务器或RS-MEC）。

2) Bandit算法。按照Bandit思想，80%的概率将任务调度到历史能耗或者时延最低的计算节点，20%的概率调度到其他计算节点。

本文测试A2C算法和基线算法在云边端调度场景下的5个关键性能指标，包括任务超时率 ω 、系统能耗 E_t^{total} 、总回报 R^{total} 、RS-MEC频率 $f_{i,t}$ 和边缘服务器频率 $f_{h,t}$ 。本文统计50 000次仿真实验数据，算法运行的平均时间为0.008 7 s/step。

能耗优先：设置 λ 的值为0.004，统计模型训练过程中以上5个关键性能指标随训练次数的变化情况。

如图7所示，能耗优先下，Rule-based算法能耗最高且不随训练次数而改变。这是因为Rule-based算法的任务调度决策取决于静态的规则，缺乏对环境的适应性。最小等待队列规则并不能反映当前计算节点的能耗效率，因此Rule-based算法的能耗始终较高且无法通过经验改进。Bandit算法通过一定的概率探索，能识别和利用历史能耗较低的节点执行任务，因此比Rule-based算法的能耗效果更好，但它只关注历史能耗，对系统当前的状态以及策略的长期回报关注不足，具备一定的局限性。A2C算法的能耗逐渐降低最终趋于稳定，且能耗最低。这是因为初期Actor和Critic都未经训练，能耗较高。随着与环境的交互进行经验收集与策略学习，调度决策越来越优，系统能耗持续下降。最后逐渐收敛至最优，在硬件能效限制下，能耗降至当前系统约束的理论下限。A2C算法能够基于当前系统状态进行决策，且最大化长期积累奖励。最终系统达到稳定的低能耗水平，体现了A2C算法的有效性与优越性。

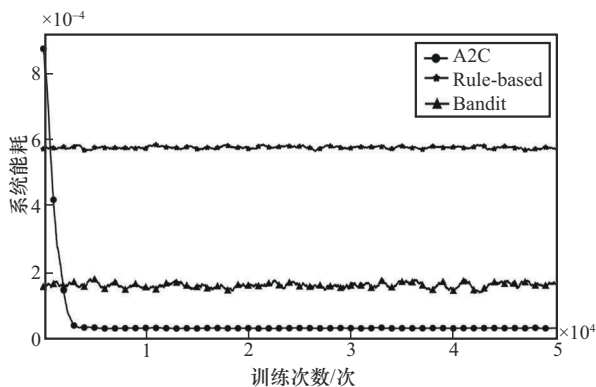


图7 能耗优先下的系统能耗变化

如图8所示，能耗优先下，A2C算法的总回报先增加后趋于稳定，总回报最高，而Bandit和Rule-based算法的总回报稳定不变，且Rule-based算法效果最差。这是因为A2C算法的Actor-Critic架构初期进行了有效的策略探索，且能考虑任务卸载的长期影响而非即时能耗，能够在复杂环境中找到最优解。Bandit算法有一定的探索机制，效果次于A2C算法。Rule-based算法基于固定规则，无法根据系统状态调整策略，忽略边缘服务器与RS-MEC在不同负载下的实际能耗差异，故其总回报性能表现最差。

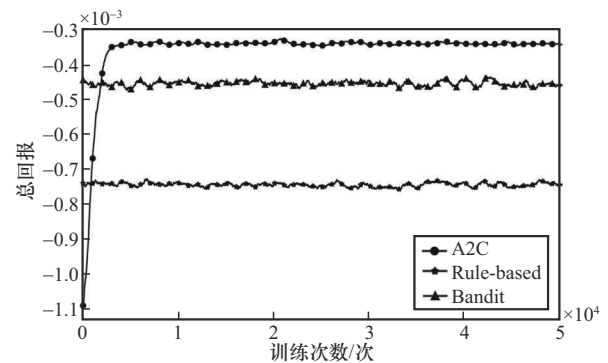


图8 能耗优先下的总回报变化

如图9和图10所示，能耗优先下，各算法对RS-MEC和边缘服务器频率影响呈现一定的差异。A2C算法下RS-MEC频率先增加后趋于稳定，边缘服务器频率先降低后稳定，说明A2C算法在初期进行了策略调度探索，并逐渐认识到RS-MEC在能耗方面的优势，增加RS-MEC频率，降低对能耗较高的边缘服务器的依赖。这导致A2C算法将更多的任务卸载到RS-MEC上，增加了任务的等待时间，所以图11中，A2C算法的任务超时率最高。边缘服务器具备较强的处理能力，故其任务队列更短，由于Rule-based算法优先选择最小任务队列的计算节点，所以将任务更多迁移到边缘服务器执行，故Rule-based算法的边缘服务器频率较高，RS-MEC频率较低，同时其任务超时率也最低。Bandit算法对于RS-MEC和边缘服务器频率影响介于A2C和Rule-based算法之间，它有较高概率选择历史能耗较低的节点，这会导致一定的任务超时，如图11所示。同时，它具有一定的概率发现当前空闲或负载较低的计算节点，所以其任务超时率低于A2C算法。

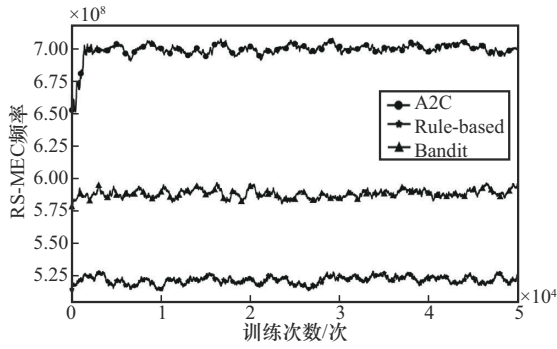


图9 能耗优先下的RS-MEC频率变化

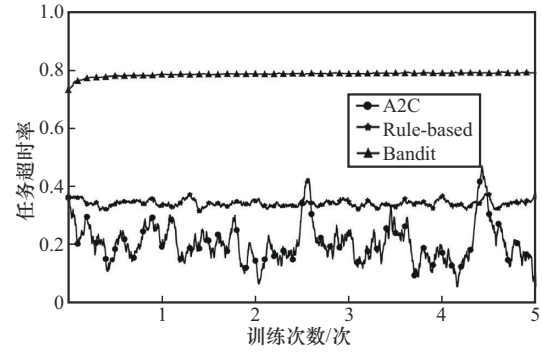


图12 时延优先下的任务超时率变化

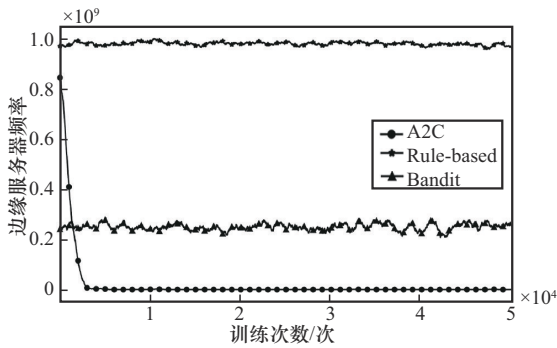


图10 能耗优先下的边缘服务器频率变化

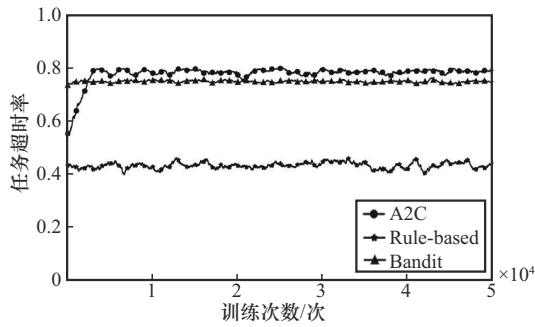


图11 能耗优先下的任务超时率变化

如图13所示，时延优先下，A2C算法的总回报最高，它不仅考虑即时时延，还考虑策略的长期回报。对应图12中的波动性，A2C算法在探索环境动态变化期间，任务卸载位置在RS-MEC和边缘服务器之间进行调整，权衡能耗和时延。Rule-based算法表现介于A2C和Bandit算法之间，任务优先在边缘服务器执行，能耗较高但任务传输时延高，故总回报性能居中。Bandit算法的总回报最低，缺乏对计算节点实时负载的感知，其任务超时率最高，总回报性能表现最差。

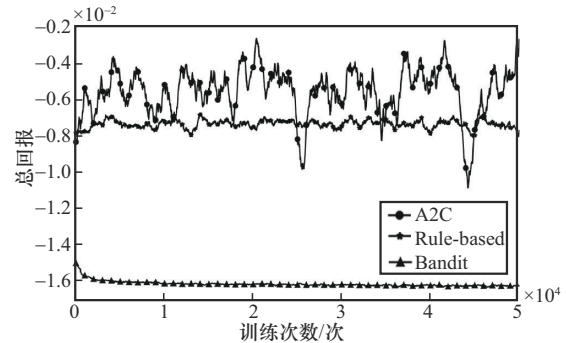


图13 时延优先下的总回报变化

时延优先。设置 λ 的值为0.02，统计模型训练过程中5个关键性能指标随训练次数的变化情况。

如图12所示，时延优化下，Bandit算法的任务超时率最高，因为其基于历史经验选择时延低的计算节点，反而导致计算节点负载增加，任务排队等待时间变长，实际时延反而增加。Rule-based算法的任务超时率居中，时延优先下优先选择处理能力强的边缘服务器，任务超时率较低。A2C算法的任务超时率最低，其任务超时率初期波动降低后保持一定程度的波动。这是因为A2C算法初期不断探索环境学习，但其总回报中也有能耗因素，故会使任务卸载在RS-MEC和边缘服务器之间进行调整，进而导致任务超时率波动，其幅度大于其他2种算法。

如图14和图15所示，时延优先下，A2C算法将任务更多在边缘服务器执行，所以边缘服务器频率较高，RS-MEC频率较低，任务迁移到边缘服务器带来了额外的开销，故如图16所示，A2C算法的系统能耗较高。Bandit算法优先选择历史时延较低的计算节点，由于RS-MEC离计算任务更近，故将任务更多卸载到RS-MEC上，其RS-MEC频率较高，当RS-MEC任务增加导致时延过高时，又会迁移到边缘服务器，故其RS-MEC和边缘服务器频率波动较大。但其有一定的探索概率，能够平衡当前策略，所以其系统能耗较低。Rule-based算法在时延优先下优先选择边缘服

务器, 故其边缘服务器频率较高。其调度规则固定, 不会产生额外的迁移成本, 故其系统能耗较低。

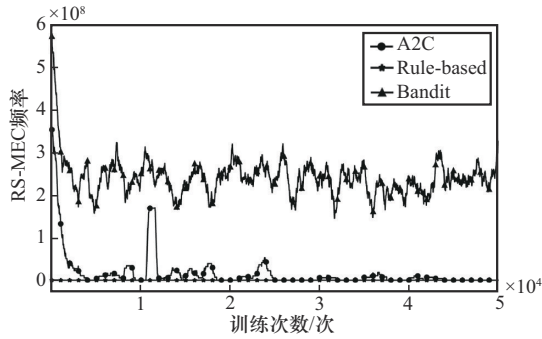


图14 时延优先下的RS-MEC频率变化

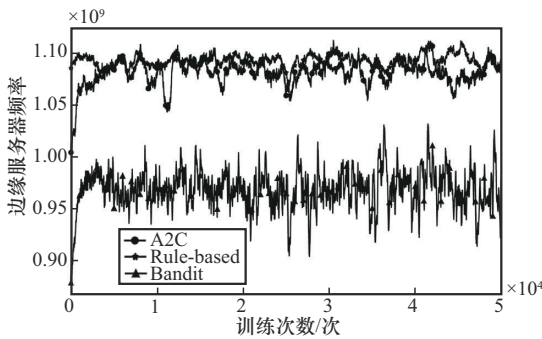


图15 时延优先下的边缘服务器频率变化

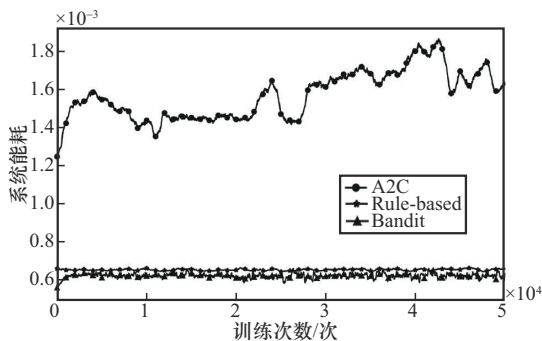


图16 时延优先下的系统能耗变化

总体来说, 在不同优化下, 本文提出的A2C算法通过Actor和Critic自适应调整策略, 训练曲线总回报在4000步之后趋于稳定, 表明该算法能在合理时间内收敛。仿真实验验证了强化学习A2C算法在能耗和时延多目标优化中的有效性和优越性, 能够适应复杂动态环境开展任务调度决策, 实现资源精细化管控。

5 部署验证

5.1 部署环境

本文方案在桐乡市车路云一体化试点区域进行

了实地部署, 对辅助驾驶下的端边调度场景进行验证, 部分部署图如图17所示。部署环境包含3个路口(云亨路-智程路路口、云亨路-隆源路路口和云亨路-姚太线路口), 每个路口配置低时延电警相机4台, 低时延卡口相机4台, 低时延鱼眼相机4台, 低时延双目雷视4台; 云亨路-隆源路路口增加4台低时延卡口相机和2台低时延鱼眼相机用于接力补盲。



(a) 相机和落地箱部署图



(b) 双目雷视部署图

图17 桐乡单路口实地部署

云亨路-智程路路口部署1台RS-MEC, 型号为Atlas500 Pro (CPU1×鲲鹏920-24核2.6 GHz, 内存4×64 GB DDR4, 硬盘2×960 GB SATA SSD, AI算力最大420TOPS INT8), 边缘服务器配置Atlas800 (型号3000, CPU2×鲲鹏920-24核2.6 GHz, 内存8×32 GB DDR4, 硬盘2×960 GB SATA SSD, AI算力最大1120TOPS INT8)。部署2台SRv6网关(飞腾D2000主机, 内存16 GB, 存储512 GB SSD, 显卡SM768)分别连接RS-MEC与边缘服务器。

实地部署拓扑如图18所示, 在区域云部署基于IPv6+的智能车联算网调度系统, 系统平台接入算力设施RS-MEC和边缘服务器, 分别部署融合感知算法, 用于处理来自摄像机和雷达的数据。系统预置调度策略, 调度模块周期性访问感知模块的感知数据, 包括算力、网络、应用信息以及部署服务

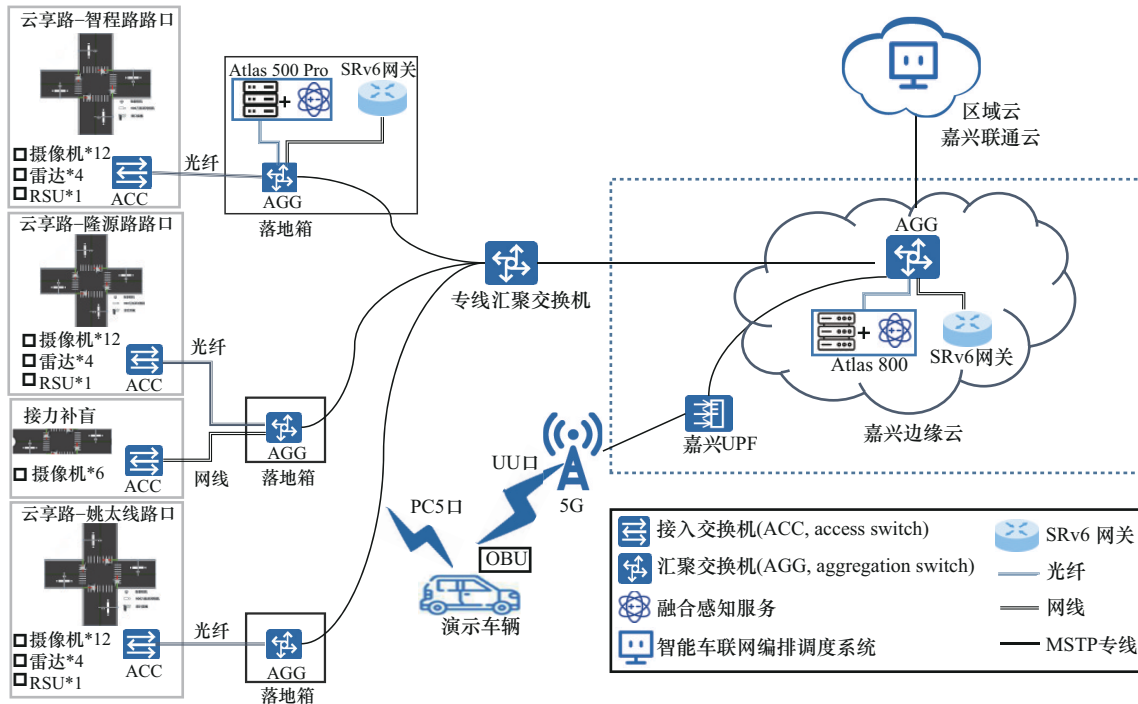


图 18 实地部署拓扑

的健康度。根据预置的调度策略，对 RS-MEC 的任务进行调度决策，确定任务数据的处理位置（RS-MEC 或边缘服务器）。

系统的调度对象为视频流和雷达数据流。针对视频流，触发调度后，调度模块分别与调度源节点和目标节点的服务进行交互，修改服务的配置文件，反馈给调度模块成功与否，并记录调度日志。视频流调度成功后，修改 SRv6 网关的推拉流服务配置，改变雷达数据流的处理位置。当资源状况满足任务执行条件后，恢复调度。此外，为避免频繁触发调度带来的 SRv6 路径重配开销问题与为降低网络抖动导致的频繁调度触发问题，本文采用双阈值触发机制与多次确认机制结合的方案进行实际部署。当 CPU 和 Mem 利用率均达到阈值时才触发调度，并且连续 3 次触发调度后执行调度。

5.2 效果分析

本文设置触发条件为：CPU 利用率大于 0.55 时或 Mem 利用率大于 0.55 时触发调度，统计不同情况下 RS-MEC 和边缘服务器在调度前后的资源利用率状况，以及整个调度流程各个阶段的时延。当 CPU 和 Mem 利用率均小于 0.4 时，恢复调度。

从图 19 和图 20 中可以看出，当 RS-MEC 上的 CPU 利用率和 Mem 利用率大于 0.55 时触发调度。触发调度后，RS-MEC 上的 CPU 利用率和 Mem 利用

率都有降低，这是由于视频流和雷达数据流的某些处理流程主要依靠 CPU 进行计算，任务迁移后，直接释放 CPU 资源，降低负载；Mem 利用率降低是因为数据缓存的需求减少，相关内存占用被释放；边缘服务器的 CPU 利用率和内存利用率小幅度增加，则是由于任务被迁移，增加了计算和缓存需求，且边缘服务器的容量较高，资源利用率增幅不明显。

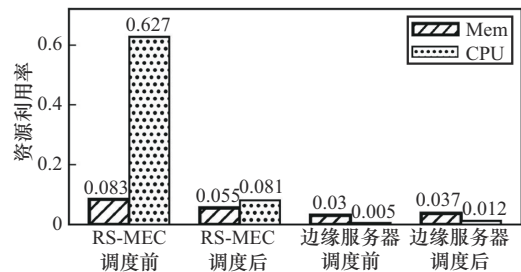


图 19 CPU 利用率大于 0.55 时触发调度服务器资源利用率

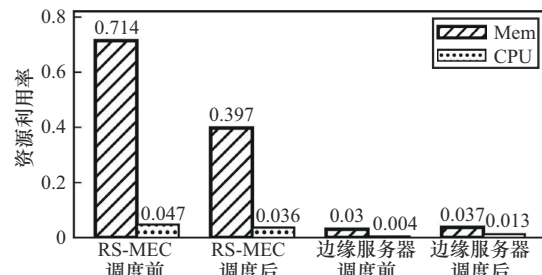


图 20 Mem 利用率大于 0.55 时触发调度服务器资源利用率

部署的融合感知服务包含多个子服务，每个子服务处理不同类型的任务且对资源的依赖程度不同。系统根据调度的触发条件开展差异化任务迁移，从而导致资源利用率的变化不同。因为 CPU 利用率达到阈值触发的调度，CPU 利用率降幅显著。系统针对计算瓶颈，迁移 CPU 密集型任务，降低 CPU 利用率；因为 Mem 利用率达到阈值触发的调度，Mem 利用率降幅显著。系统针对内存瓶颈，迁移内存密集型任务，显著降低 Mem 利用率。

综上，无论触发条件如何，调度系统都显著降低了调度源节点上的资源利用率，缓解了计算资源不足问题。通过任务调度和资源的灵活分配，也能降低算力基础设施的部署成本。

从图 21 和图 22 可以看出，调度执行过程中并行修改 RS-MEC 和边缘服务器的服务配置是执行调度的主要时延。后续，可从优化配置流程（如差异化配置同步）、调度策略改进（预测性调度提前完成节点重启）、调度触发频率改进（设置 SRv6 路径重配置的最小时间间隔、采用动态阈值调度触发策略、增加抖动抑制机制、预配置 Segment List 池与批量路径推送模式）等技术角度对调度流程进行优化。

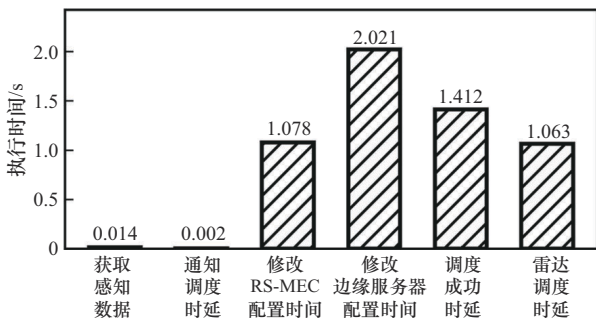


图 21 CPU 利用率大于 0.55 时触发调度各流程时延

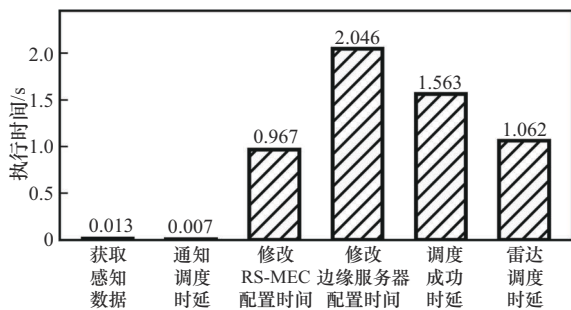


图 22 Mem 利用率大于 0.55 时触发调度各流程时延

6 结束语

本文针对车联网场景下存在的算力基础设施部

署成本高、IPv4 网络可编程性差、资源调度灵活性差等关键挑战，提出并验证了基于 IPv6+ 的智能车联算网调度方案。建立算网接入、算网感知与算网调度的层次化协同架构，基于原子编排构建场景化调度策略，实现资源的精准感知与高效调度。结合强化学习 A2C 算法，实现任务卸载位置与算力主频决策，有效平衡系统能耗与时延，并采用 SRv6 可编程路由技术实现智能选路与流量转发。仿真实验与实地部署分别验证了 A2C 算法和本文方案在典型车联网场景中的任务调度与资源动态调整的有效性和性能优越性，为车路云协同体系建设提供了兼具理论创新与工程价值的解决方案。后续本文将多区域、多复杂场景、不同网络状况等情景下的实地验证作为下一阶段的研究重点。

参考文献:

- [1] 姚惠娟, 陆璐, 段晓东. 算力感知网络架构与关键技术[J]. 中兴通讯技术, 2021, 27(3): 7-11.
YAO H J, LU L, DUAN X D. Architecture and key technologies for computing-aware networking[J]. ZTE Technology Journal, 2021, 27(3): 7-11.
- [2] 唐卓, 蒋冰婷, 张嘉鹏, 等. 算力网络调度基础理论与关键技术现状及展望[J]. 中国科学基金, 2025, 39(2): 240-249.
TANG Z, JIANG B T, ZHANG J P, et al. The foundation theory and key technologies of computing first network scheduling: present situation and prospects[J]. Bulletin of National Natural Science Foundation of China, 2025, 39(2): 240-249.
- [3] 张维庭, 孙呈慧, 王洪超, 等. 算网资源智能适配与融合调度方法[J]. 电信科学, 2023, 39(9): 12-20.
ZHANG W T, SUN C H, WANG H C, et al. Intelligent adaptation and integrated scheduling method for computing and networking resources[J]. Telecommunications Science, 2023, 39(9): 12-20.
- [4] 莫益军. 算力网络场景需求及算网融合调度机制探讨[J]. 信息通信技术, 2022, 16(2): 34-39, 84.
MO Y J. Discussion on computing network scenarios and scheduling mechanisms[J]. Information and Communications Technologies, 2022, 16(2): 34-39, 84.
- [5] TANG S J, YU Y, WANG H, et al. A survey on scheduling techniques in computing and network convergence[J]. IEEE Communications Surveys & Tutorials, 2024, 26(1): 160-195.
- [6] 曹畅, 唐雄燕. 算力网络关键技术及发展挑战分析[J]. 信息通信技术与政策, 2021(3): 6-11.
CAO C, TANG X Y. Analysis of key technologies and development challenges of computing power network[J]. Information and Communications Technology and Policy, 2021(3): 6-11.
- [7] 赵正波, 张金漫, 王艾莉, 等. 一种基于视联 AI 业务的算网调度方案[J]. 电信工程技术与标准化, 2024, 37(10): 55-61.
ZHAO Z B, ZHANG J M, WANG A L, et al. A computing network scheduling scheme based on the Internet of video AI business[J]. Tele-

- com Engineering Technics and Standardization, 2024, 37(10): 55-61.
- [8] TIAN Y, WANG Z L, YIN X, et al. Traffic engineering in partially deployed segment routing over IPv6 network with deep reinforcement learning[J]. IEEE/ACM Transactions on Networking, 2020, 28(4): 1573-1586.
- [9] KURI S K, ELDOSOUKY A, IBNKAHLA M. Performance measurement of IoT traffic through SRv6 network programming[C]//Proceedings of the 2022 IEEE International Conference on Communications Workshops (ICC Workshops). Piscataway: IEEE Press, 2022: 1-6.
- [10] LEMMI L, PULIAFITO C, VIRDIS A, et al. SRv6-based edge service continuity in 5G mobile networks[J]. Future Internet, 2024, 16(4): 138.
- [11] RAJU M R, MOTHKU S K, SOMESULA M K. MITS: mobility-aware intelligent task scheduling in vehicular fog networks[J]. IEEE Transactions on Vehicular Technology, 2024, 73(3): 3079-3093.
- [12] WANG Z Y, SUN G, SU H Y, et al. Low-latency scheduling approach for dependent tasks in MEC-enabled 5G vehicular networks[J]. IEEE Internet of Things Journal, 2024, 11(4): 6278-6289.
- [13] LI H Y, YANG Y, HAN S H, et al. Dynamic scheduling method of computing resources in Internet of vehicles based on an improved NSGAI algorithm[C]//Proceedings of the 2024 6th International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI). Piscataway: IEEE Press, 2024: 926-930.
- [14] NING Z L, HUANG J, WANG X J, et al. Mobile edge computing-enabled Internet of vehicles: toward energy-efficient scheduling[J]. IEEE Network, 2019, 33(5): 198-205.
- [15] QI Q, ZHANG L X, WANG J Y, et al. Scalable parallel task scheduling for autonomous driving using multi-task deep reinforcement learning[J]. IEEE Transactions on Vehicular Technology, 2020, 69(11): 13861-13874.
- [16] DONG J Y, GAO S T, LU H J, et al. Joint optimization of resource allocation and tasks scheduling in network slicing enabled Internet of vehicles[C]//Proceedings of the 2022 IEEE 8th International Conference on Computer and Communications (ICCC). Piscataway: IEEE Press, 2022: 552-556.
- [17] YU Z Y, TANG Y L, ZHANG L T, et al. Deep reinforcement learning based computing offloading decision and task scheduling in Internet of vehicles[C]//Proceedings of the 2021 IEEE/CIC International Conference on Communications in China (ICCC). Piscataway: IEEE Press, 2021: 1166-1171.
- [18] WANG X, HE C, JIANG W H, et al. Generative AI-based dependency-aware task offloading and resource allocation for UAV-assisted IoV[J]. IEEE Open Journal of the Communications Society, 2025, 6: 3932-3949.
- [19] LIU H Q, HUANG W Z, KIM D I, et al. Towards efficient task offloading with dependency guarantees in vehicular edge networks through distributed deep reinforcement learning[J]. IEEE Transactions on Vehicular Technology, 2024, 73(9): 13665-13681.
- [20] WU D P, LIU T, LI Z D, et al. Delay-aware edge-terminal collaboration in green Internet of vehicles: a multiagent soft actor-critic approach[J]. IEEE Transactions on Green Communications and Networking, 2023, 7(2): 1090-1102.
- [21] 宋文良. 大模型对无线传感器网络资源调度优化研究[J]. 中国宽带, 2025(5): 174-176.
- SONG W L. Research on resource scheduling optimization of wireless sensor networks based on large model[J]. China BroadBand, 2025(5): 174-176.
- [22] 金兴海. 基于大模型的云计算资源调度算法综述[J]. 电信快报, 2025(1): 15-19, 37.
- JIN X H. Review of large model-based resource scheduling algorithms for cloud computing[J]. Telecommunications Information, 2025(1): 15-19, 37.
- [23] TANG X Y, CAO C, WANG Y X, et al. Computing power network: the architecture of convergence of computing and networking towards 6G requirement[J]. China Communications, 2021, 18(2): 175-185.
- [24] 谭子航. 面向 6G 网络的跨域资源编排和分配技术研究[D]. 西安: 西安电子科技大学, 2024.
- TAN Z H. Research on cross-domain resource orchestration and allocation for 6G networks[D]. Xi'an: Xidian University, 2024.
- [25] 中兴通讯股份有限公司. SRv6 技术白皮书[R]. 2020.
- ZTE Corporation. SRv6 technical white paper[R]. 2020.
- [26] 雷鹏. 基于 A2C 算法考虑投资风险的股票交易策略研究[D]. 成都: 西南财经大学, 2024.
- LEI P. Research on stock trading strategies considering investment risk based on A2C algorithm[D]. Chengdu: Southwestern University of Finance and Economics, 2024.
- [27] MAO Y Y, ZHANG J, SONG S H, et al. Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems[J]. IEEE Transactions on Wireless Communications, 2017, 16(9): 5994-6009.

[作者简介]



于朝阳 (1996-), 女, 河北沧州人, 博士, 联通智网科技股份有限公司车辆智能网联研究院工程师, 主要研究方向为边缘计算、深度强化学习、车联网、任务调度及资源分配。

辛亮 (1987-), 男, 江西宜春人, 北京邮电大学博士生, 联通智网科技股份有限公司车辆智能网联研究院高级工程师, 主要研究方向为车联网算力网络、5G-A/6G 网络、车联网创新应用场景等。

杨洋 (1990-), 女, 吉林松原人, 中国联合网络通信有限公司研究院工程师, 主要研究方向为算力网络、云网融合/云计算、算网编排调度技术等。

周光涛 (1978-), 男, 吉林松原人, 博士, 联通智网科技股份有限公司车辆智能网联研究院教授级高级工程师, 主要研究方向为新型车联网架构、5G 与边缘计算融合、车路云一体化等。

马楠 (1979-), 男, 陕西旬阳人, 博士, 北京邮电大学教授, 主要研究方向为移动通信理论与测试技术、语义通信。