计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
63(8): 1917−1933, 2026

本期推荐

广域联盟算力网的服务质量反馈控制机制

周钰雯[1] 任棒棒[1] 何 瑞[1] 郭得科[2]
[1](国防科技大学信息系统工程全国重点实验室 长沙 410073)
[2](中山大学计算机学院 广州 510006)
(zhouyuwen23@nudt.edu.cn)

Quality of Service Feedback Control Mechanism in Wide-Area Computility Federation

Zhou Yuwen[1], Ren Bangbang[1], He Rui[1], and Guo Deke[2]
[1](*National Key Laboratory of Information Systems Engineering*, *National University of Defense Technology*, *Changsha* 410073)
[2](*School of Computer Science and Engineering*, *Sun Yat-sen University*, *Guangzhou* 510006)

Abstract Computility federations coordinate heterogeneous resources across independent providers while preserving administrative autonomy. This autonomy limits federation-level visibility, forcing gateways to route from coarse-grained and delayed performance summaries rather than real-time internal states. Under such weak and lagged observability, deadline-miss risk for latency-critical tasks can become structurally concentrated in a small number of domains and time windows, even when federation-wide averages remain moderate. To address this problem, we propose per-domain SLO feedback (PDSF), a low-dimensional closed-loop mechanism that updates bounded routing tolls from window-level summaries and biases snapshot-driven self-routing without modifying domain-local schedulers. Theoretical analysis shows that, under a weak monotonicity assumption on routing response, PDSF exhibits boundedness, forgetting, and local stability under delayed feedback. Trace-driven experiments on a multi-domain emulator using Google cluster workloads show that, compared with deployable baselines, PDSF significantly alleviates upper-tail overload severity, reduces worst-domain risk exposure, and protects latency-critical traffic without noticeable degradation to best-effort traffic. Sensitivity analysis further shows that snapshot staleness and routing concentration jointly determine the mechanism's operating boundary and clarifies how stale measurements and aggressive routing response amplify risk while reducing the effectiveness of low-dimensional feedback.

Key words computility federation; cross-domain routing; tail latency; delayed feedback; overload control

摘 要 联盟算力网(computility federation)旨在协调跨域异构计算资源,同时保留各独立提供商的管理自治权。然而,这种自治性也限制了联盟层的全局可观测性,使各域网关在进行路由决策时无法获得实时域内状态,只能依赖粗粒度且存在滞后的性能摘要。在这种弱可观测和状态滞后的条件下,时延敏感型任务的超时风险容易在少数域和少数窗口内发生结构性聚集。即使联盟整体平均负载仍处于合理水平,局部域也可能在偶发过载窗口中承受显著升高的风险暴露。针对这一问题,提出一种域级 SLO 反馈机制 PDSF。该机制是一种低维闭环调节方法,无需修改域内本地调度器,而是利用窗口级统计摘要动态更新时延敏感型任务的路由代价,从而对基于状态快照的自路由过程施加偏置调节。理论分析表明,在

收稿日期: 2026-03-04; **修回日期**: 2026-04-27
基金项目: 广东省基础与应用基础研究重大项目(2026B0303000007); 国家自然科学基金项目(U23B2004)
This work was supported by the Guangdong Major Project of Basic Research (2026B0303000007) and National Natural Science Foundation of China (U23B2004).
通信作者: 郭得科(guodk@mail.sysu.edu.cn)

路由响应满足弱单调性的条件下,PDSF具有有界性、遗忘性以及延迟反馈下的局部稳定性。基于Google集群工作负载轨迹的多域仿真实验表明,与现有可部署基线相比,PDSF能够显著缓解长尾过载,降低最差域风险暴露,并在保护时延敏感型任务的同时避免尽力而为型任务性能出现明显退化。进一步的敏感性分析表明,快照滞后与路由集中度共同决定了该机制的适用边界,并表明了测量信息陈旧性与路由反应性如何放大系统风险、压缩低维反馈调节的有效空间。

关键词 联盟算力网;跨域路由;长尾时延;延迟反馈;过载控制

中图法分类号 TP393; TP338.8

DOI: 10.7544/issn1000-1239.202660112 **CSTR:** 32373.14.issn1000-1239.202660112

联盟算力网(computility federation)旨在将隶属于不同管理域、地理分布的计算、网络及存储资源进行互联[1-3]。天空计算(sky computing)[4]和云际(Intercloud)计算架构[5]等相关愿景也强调在保留资源提供商本地自治权的同时,实现对异构算力的统一访问。因此可以把联盟算力网视作统一的控制面,它在不改变资源提供商侧调度策略的前提下,对外暴露统一的编排接口[4-6]。在此类场景中,联盟级控制器通常仅能观测到粗粒度的域网关统计摘要。这些摘要包括在监控窗口内聚合的各类任务的到达率、截止期违例(deadline misses)数量以及长尾时延百分位等。后续将“任务响应时间超过其所属类别截止期”的事件记为一次截止期违例,并将相应的截止期违例比例简称为超时率。

受限于各域的管理边界,联盟算力网常采用分层控制架构,这与当前已有的联盟资源管理器及多域调度器类似[7-9]。如图1所示,资源被划分为若干域,每个域由一个域网关(domain gateway, DG)作为面向联盟的入口。每个网关利用联盟级信号及近期的窗口摘要为每个任务选择目标域,并将任务转发至目标域,由该域的本地调度器执行后续的调度。网关同时按域和类别聚合统计摘要,并将其上报至联盟编排器(federation orchestration, FO)。FO仅能在时间窗口结束时下发低维的域级路由代价(toll)作为联盟级信号来引导DG的后续决策,而细粒度的逐任务路由转发仍由DG独立完成,且域内排队与任务调度策略对联盟不可见[7,9]。此外,系统中的任务被划分为不同的服务类别以匹配相应的服务质量(quality of service, QoS)目标,主要包括对时延要求极高的时延敏感(latency-critical, LC)型任务,以及对时延不敏感的尽力而为(best-effort, BE)型任务。

这种责任分离机制在提升可扩展性并保障提供

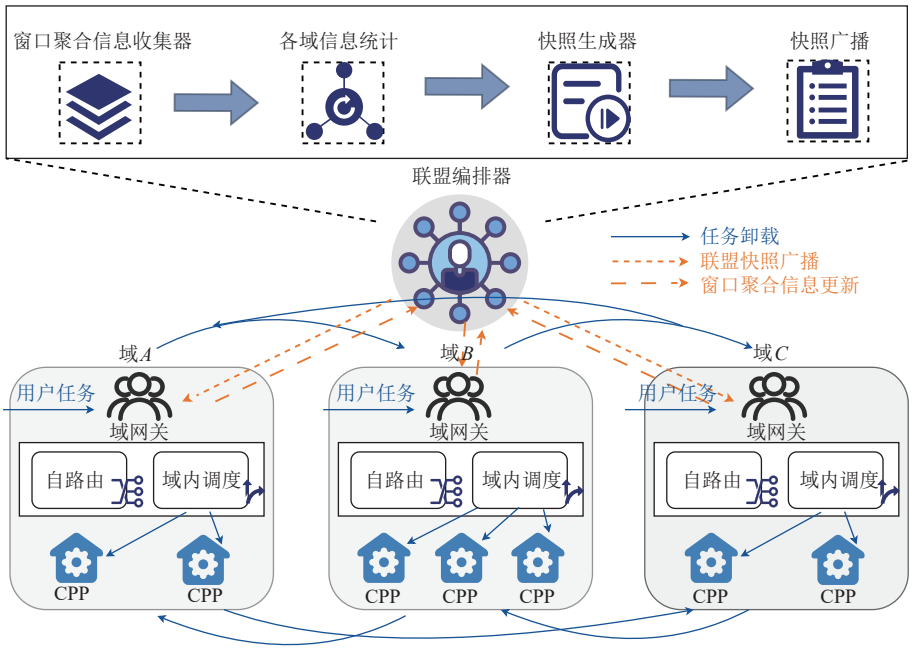


Fig. 1 Hierarchical control architecture of computility federation

图1 联盟算力网的分层控制架构

商自治权的同时，也增加了联盟层面保障严格长尾时延目标的难度。FO 仅能接收经延迟和聚合处理的测量数据，并通过缓慢更新各域选择时的域级权重来干预路由决策。与此同时，DG 在执行逐任务路由选择时，同样依赖于滞后的状态快照。在这种双重滞后下，时延敏感型任务的流量容易持续涌向历史上看似状态较优的少数域，从而使超时风险在局部发生结构性聚集。也就是说，即便联盟整体的平均指标看似稳定，局部高风险域仍可能反复出现[10]。进一步地，长尾时延对瞬时拥塞十分敏感，短暂的队列积压就可能迅速推高高分位时延，并诱发任务超时，因此会进一步放大这种局部风险[11]。同时，经典反馈控制理论指出，针对噪声测量数据采取激进的响应策略可能在大规模系统中诱发震荡[12]。因此，在窗口级反馈、测量滞后以及域内状态不透明的共同约束下，联盟层既难以获得支撑全局最优决策所需的充分信息，也不适合通过高增益控制对路由进行剧烈干预。基于此，本文的目标并非追求理想化的全局可行性或最优性，而是在现实接口约束下，利用低维、缓变的反馈信号重塑时延敏感任务的超时风险在各域之间的分布形态，抑制其在少数域内的结构性聚集。

基于上述分析，本文提出域级 SLO 反馈机制 PDSF(per-domain SLO feedback)，以在窗口级滞后反馈和域内状态不透明的条件下抑制时延敏感型任务风险的局部聚集。PDSF 以近期观测到的超时率相对于目标值的偏差为输入，由 FO 在窗口边界为各域平滑更新有界通行代价。当某域持续承受较高的超时压力时，其通行代价增加，进而提高该域在 DG 视角下的等效路由开销，抑制后续流量在滞后快照驱动下继续向该域汇聚；当压力缓解时，该代价则逐步回落。由于整个更新过程仅依赖窗口级摘要，而不依赖域内队列状态或本地调度细节，PDSF 能够在不修改域侧调度器的前提下实现联盟级干预。从理论上看，这一机制与背压方法利用可观测拥塞信号调节流量的思想相一致，从而为延迟反馈下的低维风险控制提供了可解释的依据[13-14]。

本文的主要贡献总结如下：

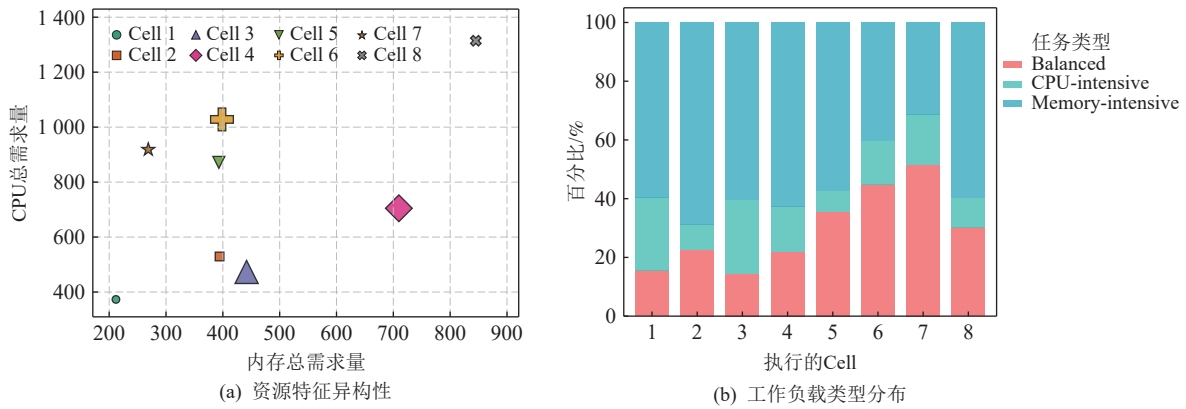
1)面向窗口级可观测、快照滞后且域内状态不可见的联盟场景，形式化建模了基于快照引导的跨域路由问题，并提出了刻画局部过载与最差域暴露的结构性风险指标。

2)提出域级 SLO 反馈机制 PDSF。该机制通过窗口级统计摘要动态更新有界路由代价，在不干预域内调度的条件下抑制时延敏感型任务风险的结构性聚集，并给出了延迟反馈条件下的有界性、遗忘性和局部稳定性分析。

3)基于轨迹驱动的多域仿真评估了所提方法，并通过与可部署基线及全局信息理想方案的对比，揭示了快照滞后和路由集中度对系统效能及方法适用范围的影响。

1 研究动机

联盟算力网聚合了来自多个独立提供商的资源。其显著特征在于供给侧和需求侧的异构性。在供给侧，各域在硬件能力、背景负载及运行条件上存在差异；在需求侧，工作负载并非同质，包含资源特征各异的任务。分析 Google 集群轨迹可知，任务在 CPU 和内存需求上表现出显著异构性，且其到达过程具有明显的动态波动特征[15]。图 2(a)中各个标记的大小表示任务到达数量的规模，直观展示了这种异构性。图 2(b)进一步表明，不同计算集群所处理的任务类型分布存在显著差异，呈现明显的非均匀特性，



(a) 资源特征异构性　　(b) 工作负载类型分布

Fig. 2 Structural heterogeneity in Google cluster trace

图 2 Google 集群轨迹中的结构异构性

对任务具体的聚类细节将在第 5 节详述。结合域的异构性, 这种偏斜意味着跨域性能可能因任务特征而异。这种差异在域高负载运行时尤为显著, 并最终体现在长尾时延的恶化和任务超时的加剧上。

异构性使得跨域协调具有价值, 但联盟受到自治和治理要求的约束。在许多联盟环境中, 细粒度的内部遥测数据对联盟控制层完全屏蔽, 以防止泄露提供商敏感的运行模式或打破其管理边界[4,6,16]。相反, FO 只能依赖 DG 周期性上报的粗粒度窗口聚合信息来评估服务目标的达成情况。这种受限的可观测性既是满足系统实际运行约束的必然选择, 也契合了反馈控制理论对系统稳定性的要求。先前的研究表明, 直接利用高频噪声测量驱动决策可能会放大波动, 并在大规模系统中诱发震荡行为[12], 因此 FO 采用离散周期更新聚合摘要, 而非对每个任务作出反应。

由此产生的时间尺度失配, 会显著削弱时延敏感型任务的性能保障。在每个监控窗口内, DG 根据上一窗口的长尾时延快照为新到达任务选择目标域。如果某个域在历史快照中表现较优, 多个入口网关便可能在下一窗口内同时将大量时延敏感型任务引向该域。在高并发条件下, 这种同步汇聚会迅速推高该域的排队压力, 使其长尾时延在窗口内部急剧恶化, 并进一步诱发大量任务超时。然而, 窗口结束后生成的统计摘要只能反映已经发生的结果, 无法刻画拥塞形成的具体时机及性能劣化的演进过程。与此同时, FO 接收到的也是带有滞后的聚合信息, 因此难以及时纠正这种由旧快照驱动的流量偏移。结果是, 即便从联盟整体看路由份额似乎大体均衡, 时延敏感型任务的超时风险仍可能持续集中在少数域内, 形成明显的结构性聚集。

图 3 直观展示了滞后窗口摘要如何诱发局部过载聚集。对于图中的每个域, 左侧圆柱体表示 DG 在当前窗口进行路由决策时所依据的过时快照, 即 $T-\Delta t$时刻上报的严格类第 95 百分位时延; 右侧圆柱体表示在当前窗口 T 内, 经过实际路由转发与队列积压后测得的严格类第 95 百分位时延。两者之间的差异表明, 历史快照中的“优选域”在真实执行过程中可能迅速转变为局部拥塞域。由于路由决策依赖滞后快照且缺乏及时校正信号, 多个 DG 容易同时将流量引向同一历史上表现较优的域, 从而使该域内的排队压力和任务超时在窗口内部快速累积。为了抑制这种由旧快照驱动的局部聚集, 本文提出 PDSF 机制。该机制利用窗口级的任务到达量和实际超时统计量, 经平滑处理后动态更新区分服务类别的域级路由代价。当某个域持续承受较高的超时压力时, 其通行代价相应增加, 从而提高该域在 DG 视角下的等效路由开销, 并引导后续窗口中的部分时延敏感型任务分流至其他域。整个过程仅依赖窗口级摘要, 无需侵入或修改底层域内调度器。

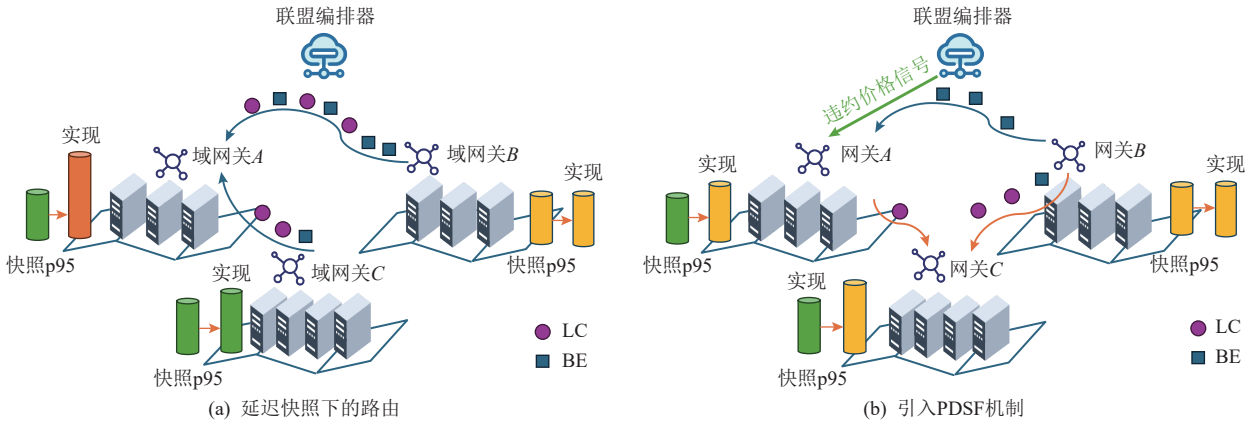


Fig. 3 Illustration of window-level observability and its impact on latency-sensitive task tail latency in a three-domain federation

图 3 窗口级可观测性及其对三域联盟中时延敏感型任务长尾时延的影响示意图

2 相关工作

基于快照的路由与负载均衡通常依赖粗粒度且可能存在噪声和滞后的测量信号。当决策者依据陈旧信息进行响应时, 闭环效应容易诱发羊群效应和瞬态过载, 即使系统的长期平均负载仍处于适中水平。文献 [17] 对集群负载均衡中的延迟反馈进行了明确建模, 并分析了其对系统性能和稳定性的影响。相关系统研究进一步指出, 负载均衡器需要结合路由路径上的带内(in-band)信号来感知服务状态, 同时也揭示了依赖不完整反馈所带来的局限性[18]。此

外，关于旧信息价值及其局限性的研究表明，信息的“滞后性”本身应被视为影响决策效果的关键因素，而不能简单看作可忽略的系统噪声[19]。类似地，全局负载均衡领域的研究也反复表明，实际路由行为往往受制于测量链路与系统接口，而非建立在全知状态之上。

经典研究表明，引入有限随机选择能够缓解大规模系统中的排队不平衡，并抑制流量向少数节点过度集中的现象[20-22]。这一思想与本文关注的场景密切相关，因为当多个入口共享同一滞后快照时，纯粹的贪婪选择往往会将原本细微的测量差异放大为同步的流量偏移。在异构入口环境下，Softmin 路由是兼顾时延感知与代价选择的典型代表，其对应的多项 Logit 映射可由随机效用模型解释，即每个入口在感知成本时都会受到微小特异扰动的影响[23-24]。更一般地，Softmin 路由、确定性最小成本路由以及有限选择路由都具有单调成本响应特性，即目标域的成本越高，其被选中的概率越低。

在控制层面，反馈控制已被广泛用于在噪声测量和时滞条件下调节系统性能[12,25]，AIMD 类规则则为分布式自适应提供了典型基线[26]。此外，定价和虚拟队列视角给出了另一种互补解释，即利用低维影子价格（shadow price）在不暴露内部状态的前提下引导去中心化决策[14,27]。本文采用的通行代价信号与这一思路相一致，但其更新并不依赖细粒度内部状态，而是依据窗口级 SLO 超额量进行调整，并仅作为与域自治约束相兼容的路由偏置使用。

长尾事件往往主导大规模服务中的用户感知可靠性，罕见的过载窗口也可能进一步引发任务超时和运行事故[11]。业界广泛采用的 SLO 与错误预算实践，正是通过将关注重点从平均性能转向高负载条件下的风险管控，对这一理念进行了系统化表达[28]。近年来，相关系统研究主要通过准入控制、节流和降级等机制来缓解单一管理域内的过载问题，通常也都围绕明确的 SLO 目标展开[29-34]。然而，这类方法大多建立在较强的控制与观测能力之上，通常假设系统能够获得丰富的服务端接口、细粒度遥测信息，或直接干预域内调度过程。针对跨云与联盟云环境下的任务调度问题，已有研究对异构资源、共享机制及任务调度策略进行了系统梳理，并指出联盟化环境中的资源异构性与跨域协调会显著增加调度复杂度[35]。相比之下，本文考虑的是一种更受约束的联盟环境，即域内调度器保持不透明，联盟层只能获得按窗口聚合且存在滞后的域级摘要，控制动作也只能以域级路由代价的形式在窗口边界缓慢更新，而不能直接介入逐任务放置或本地调度。在这种设定下，本文关注的也不再是传统意义上的平均性能优化或吞吐量最大化，而是如何利用与域自治相兼容的低维反馈信号，抑制时延敏感型任务风险在少数域内的结构性聚集。因而，PDSF 更适合被理解为面向黑盒自治域的联盟级反馈机制，而非对现有单域过载控制或影子价格方法的直接移植。

3 系统模型与问题描述

为便于阅读，本文将系统模型、控制变量及评价指标中出现的主要符号汇总在表 1 中。

3.1 异构联盟域

联盟算力网通过统一接口协调多个独立提供商拥有的资源，而这一接口同时体现了域间的管理边界，并限制了联盟层的可观测与控制范围。系统由一个 FO 以及一组域 $\mathcal{D}=\{d_1,\cdots,d_M\}$ 组成。其中每个域 d_j 是一个由 1 个或多个提供商运营的自治调度区域，并通过域网关 g_j 与联盟交互。网关部署在域边界，负责接收任务、执行跨域转发并向联盟上报状态。当任务到达网关时，网关依据联盟广播及最新可用的窗口摘要选择目标域，再将任务转发至相应的域内调度器。任务执行完成后，网关按类别汇总并上报粗粒度统计摘要，如任务量、截止期违例数及长尾时延快照等。相比之下，域内的排队、优先级划分及任务放置等机制由提供商本地维护，对联盟层不可见。由于不同域在硬件能力、运行条件及工作负载混合上普遍存在异构性，这种受限接口所带来的信息缺口会进一步放大其影响，因此本文将每个域 d_j 统一抽象为一个黑盒执行系统，其内部行为由不对外公开的底层属性决定。具体而言，本文用 $\boldsymbol{\phi}_j=\left(\phi_j^{\mathrm{cpu}},\phi_j^{\mathrm{mem}}\right)$ 表示域 d_j 的资源配置，用于刻画其处理 CPU 密集型与内存密集型任务的相对服务能力。同时，用 L_j^{base} 表示域 d_j 的基准时延，用于表征单次任务处理中与具体负载无关的固定开销，如域内网络时延等。这些信息均属于域内私有属性，对 FO 或其他域不可见。任务随时间到达 DG，并被划分为不同的服务类别。本文考虑 2 类代表性类别，即具有严格延迟要求的时延敏感型任务 LC，以及要求较宽松的尽力而为型任务 BE。记类别集合为 $\mathcal{S}=\{\mathrm{LC},\mathrm{BE}\}$。对任一到达任务 $\mathcal{T}_k$，本文将其表示为六元组 $\mathcal{T}_k=(t_k^{\mathrm{arr}},T_k^{\mathrm{ideal}},r_k^{\mathrm{cpu}},r_k^{\mathrm{mem}},\psi_k,s_k)$。其中 t_k^{arr} 为到达时间，T_k^{ideal} 为基准参考机器上的理想处理时间，r_k^{cpu} 和 r_k^{mem} 表示

Table 1 Description of Main Symbols

表 1 主要符号说明

符号	含义
$\mathcal{D}=\{d_1,\cdots,d_M\}$	联盟中的域集合
d_j	第j个域
g_j	第j个入口域网关
$\mathcal{S}=\{\mathrm{LC},\mathrm{BE}\}$	服务类别集合
LC	时延敏感型任务
BE	尽力而为型任务
$\mathcal{T}_k$	第k个任务的六元组表示
$\boldsymbol{\phi}_j=(\phi_j^{\mathrm{cpu}},\phi_j^{\mathrm{mem}})$	域d_j的资源配置
L_j^{base}	域d_j的基准时延或固定开销
θ_s	类别s的目标超时率
t	监控窗口索引
$A_{j,s}^{(t)}$	窗口t内域d_j上类别s的任务到达量
$V_{j,s}^{(t)}$	窗口t内域d_j上类别s的截止期违例数
$\hat{L}_{j,s}^{(t)}$	窗口t内域d_j上类别s的长尾时延快照
$\hat{v}_{j,s}^{(t)}$	窗口t内域d_j上类别s的观测超时率
$\boldsymbol{s}_j^{(t)}$	域d_j在窗口t上报给 FO 的摘要状态
$\mathcal{S}_{\mathrm{ctl}}\subseteq\mathcal{S}$	应用路由代价控制的受控类别集合
$\tau_{j,s}^{(t)}$	窗口t内域d_j针对类别s的通行代价
ℓ	快照滞后窗口数
ℓ_{ctl}	反馈延迟窗口数
$R_{k,j}$	入口网关到候选域d_j的往返网络时延
$C_{j,s}^{(t)}$	窗口t内类别s到域d_j的广义路由代价
$p_{j\mid s,t}$	窗口t内类别s任务被路由到域d_j的份额
β_s	类别s的 Softmin 逆温度参数
γ	泄漏因子
$E_{j,s}^{(t)}$	归一化超额违例量
σ_s	类别s的压缩参数
$e_{j,s}^{(t)}$	有界压缩后的惩罚信号
$w_{j,s}^{(t)}$	样本量加权因子
a_0	样本量平滑参数
η_s	类别s的步长参数
$\tau_{\min},\tau_{\max}$	通行代价的下界与上界
$h(\tau)$	条件期望响应函数
τ^*	响应函数对应的不动点
$S^{(t)}$	窗口t的长尾汇聚严重程度
W	最差域风险暴露水平
$A_{\min}$	最小到达量过滤阈值

CPU 和内存需求，ψ_k表示粗粒度的任务特征，而$s_k\in\mathcal{S}$表示任务所属类别。由于域异构性$\boldsymbol{\phi}_j$会影响不同类型任务的执行表现，这一任务表示能够为后续的执行时延建模提供必要条件。在许多集群系统中，尽管时延敏感型任务比尽力而为型任务具有更高优先级，但尽力而为型任务对底层资源的竞争依然会严重影响前者的长尾时延[36-39]。因此，即便本文的主要目标是保护时延敏感型任务，但仍需维护分类型的统计摘要。对于每个任务类别$s_k\in\mathcal{S}$，联盟指定单任务截止期函数$D_s(T_k^{\mathrm{ideal}})$及类级目标违例率$\theta_s\in[0,1]$。其中，时延敏感型任务对应的截止期函数比尽力而为型任务更严格，而θ_s表示系统对类别s任务超时比例的容忍上界。需要说明的是，在突发到达、非平稳性及域内黑盒动态等因素的共同作用下，实际观测到的超时率可能暂时超过θ_s，因此本文将其视为指导性的参考基准，而非绝对的硬性约束。

3.2 时延分解

FO 最终只能观测到任务完成后的端到端时延，而产生这一结果的具体机制则封装在域内部。为便于后续分析，本文将任务时延分解为域内处理时延和跨域传播时延 2 部分。对于在域d_j上执行的任务$\mathcal{T}_k$，其域内处理时延可解耦为$L_{k,j}^{\mathrm{dom}}=L_j^{\mathrm{base}}+W_{k,j}+S_{k,j}+Z_{k,j}$。其中，$L_j^{\mathrm{base}}$表示域内与负载无关的固定开销，如域内网络延迟；$W_{k,j}$表示任务在域$d_j$内由排队和本地调度策略引起的等待时间；$S_{k,j}$表示任务的服务时间，它由任务特征$(T_k^{\mathrm{ideal}},r_k^{\mathrm{cpu}},r_k^{\mathrm{mem}},\psi_k)$及域执行配置$\boldsymbol{\phi}_j$共同决定；$Z_{k,j}$则用于吸收其他可能的短期波动造成的时延。需要说明的是，这种分解仅服务于理论分析与建模推导，并不意味着这些分量在联盟层可被单独观测或直接控制。当任务发生跨域卸载时会产生传播延迟，并构成入口网关观测到的端到端时延的一部分。设g_k为任务$\mathcal{T}_k$进入联盟时所在的域网关，$R_{k,j}\geqslant 0$表示g_k到目标域d_j之间的往返网络延迟，则任务的完成时延为

$$L_{k,d_{k\to j}}^{\mathrm{e2e}}=R_{k,j}+L_{k,j}^{\mathrm{dom}}\ 。\tag{1}$$

因此，在联盟层看来，内部队列长度、本地服务规则以及形成$W_{k,j}$和$Z_{k,j}$的具体扰动条件都是不可见的，联盟能够利用的只有任务完成后体现出来的聚合结果，而非这些结果背后的细粒度形成过程。

3.3 窗口级可观测性与 FO 及 DG 交互

鉴于域内状态对联盟层不可见，且路由决策与任务完成在时间上并不同步，反馈至 FO 的信息需按监控窗口进行聚合。为此，本文将时间被划分为索引为$t=1,2,\cdots$的离散窗口，每个窗口具有固定的时长。在窗口内部，域网关针对到达任务逐一做出路由决策，并将其转发至选定域的本地调度器；窗口结束后，各网关再对所属域在该窗口内的执行结果进行汇总，

并仅向 FO 上报粗粒度统计摘要。具体而言，对于每个服务类别$s\in\mathcal{S}$，域网关g_j记录在窗口t内由域d_j完成的s类任务数，记为$A_{j,s}^{t}$，以及其中的截止期违例数，记为$V_{j,s}^{t}$。当任务的测量端到端时延超过其所属类别的截止期时，即计为 1 次截止期违例。若任务发生跨域卸载，则该时延包含传播项$R_{o,j}$。此外，网关g_j同时还会报告一个长尾时延快照$\hat{L}_{j,s}^{t}$，作为对外发布的域级执行状态信号。为进一步说明联盟中各部分的信息与控制边界，表 2 总结了 FO、DG 与域内调度器三者在可见信息以及可执行动作上的差异。相应地，窗口t内域d_j上类别s的观测超时率定义为

$$\hat{v}_{j,s}^{t}=\frac{V_{j,s}^{t}}{A_{j,s}^{(t)}+\varepsilon}, \tag{2}$$

其中$\varepsilon>0$用于避免窗口内无s类任务完成时出现分母为 0。窗口结束时，域d_j向 FO 上报状态元组

$$\boldsymbol{s}_j^{t}=(A_{j,s}^{t},V_{j,s}^{t},\hat{L}_{j,s}^{t})。 \tag{3}$$

该元组包含了域、类别及窗口 3 个维度的信息，也是联盟层关于域d_j的唯一可用观测。FO 基于上述按类别聚合的有限摘要，在联盟层实施差异化的动态路由调节。设$\mathcal{S}_{\mathrm{ctl}}\subseteq\mathcal{S}$为应用路由代价控制的任务类别集合。FO 为各域维护非负通行代价$\tau_{j,s}^{t}\geqslant 0$。除非另有说明，对于$s\in\mathcal{S}\backslash\mathcal{S}_{\mathrm{ctl}}$的任务，保持$\tau_{j,s}^{t}\equiv 0$。这些通行代价在窗口边界利用可用摘要进行更新，并随窗口级信息一并广播至各个域。本文的主要评估设定取$\mathcal{S}_{\mathrm{ctl}}=\{\mathrm{LC}\}$。

Table 2 Information Visibility and Control Boundaries

表 2 信息可见性与控制边界

层级	可见信息	可执行动作
FO	窗口级到达量、违例数、长尾时延快照、历史路由代价	窗口边界执行域级路由代价
DG	最新可用快照、联盟广播路由代价、本地入口任务属性、跨域 RTT	逐任务选择目标域
域内调度器	本域本地状态、任务队列、资源视图	本地排队与调度

3.4 路由模型

在单个监控窗口内，本文将最新可用的长尾时延快照及通行代价视为常量。设$\ell\in\{0,1,2,\cdots\}$为以窗口数度量的快照滞后，则窗口t内可用的域级长尾时延快照定义为$\tilde{L}_{j,s}^{t}=\hat{L}_{j,s}^{t-\ell}$。设$g_k$为任务进入联盟时所在的域网关，$R_{k,j}$表示从网关到候选目标域$d_j$的往返网络延迟。对于在窗口$t$内到达网关$g_k$的$s$类任务，网关为每个候选域$d_j$计算一个广义代价

$$C_{j,s}^{t}=w_{\mathrm{lat}}^{s}\tilde{L}_{j,s}^{t}+w_{\tau}^{s}\tau_{j,s}^{t}+w_{\mathrm{net}}R_{k,j}, \tag{4}$$

其中$w_{\mathrm{lat}}^{s}\geqslant 0$和$w_{\tau}^{s}\geqslant 0$是与任务类别相关的权重参数，$w_{\mathrm{net}}\geqslant 0$为网络时延项的权重。对于非受控类别，通行代价项保持固定且通常设为 0，等价于对$s\in\mathcal{S}\backslash\mathcal{S}_{\mathrm{ctl}}$设定$w_{\tau}^{s}=0$。显然，$C_{j,s}^{t}$越小，表明域$d_j$对$s$类任务越具吸引力。路由策略进一步将代价集合$\{C_{j,s}^{t}\}_{j\in\mathcal{D}}$映射为入口网关$g_k$面向类别$s$的流量路由份额。本文对该映射施加一个基本的单调性要求，即对于任意$g_k,s,$和d_j，若目标域的代价$C_{j,s}^{t}$上升而其他域的代价保持不变，则分配给该域的流量份额不应增加。由于路由代价$\tau_{j,s}^{t}$以非负系数进入$C_{j,s}^{t}$，因此增加$\tau_{j,s}^{t}$不会导致任何受控类别在d_j上的流量份额上升。确定性的最小代价路由、Softmin 规则以及有限选择路由都满足这一性质。一种典型的实现是采用基于多项 Logit（multinomial-logit）的 Softmin 规则：

$$p_{j|s,t}=\frac{\exp\left(-\beta_s C_{j,s}^{t}\right)}{\sum_{m\in\mathcal{D}}\exp\left(-\beta_s C_{m,s}^{t}\right)}, \tag{5}$$

其中$p_{j|s,t}$表示类别s任务在窗口t被路由到域d_j的概率。在随机效用理论中，该映射可解释为一种带有微小特异性扰动的代价感知选择机制[24]。参数$\beta_s>0$为控制路由的集中度，即β_s较大时，路由行为更接近确定性最小代价选择，而β_s较小时，流量分配更加平滑，从而有助于缓解因快照噪声引发的路由剧烈震荡。

3.5 联盟级控制目标

FO 运行于各自治域之上，并仅通过窗口级接口与域网关交互。逐任务路由由网关执行，域内调度则保持本地化且对联盟层不可见。因此，FO 既不能为单个任务直接指定执行目标，也无法观测域内的队列状态和服务规则，其唯一的调节通道是作用于慢时间尺度上的域级通行代价信号。该信号通过对第 3.4 节所述时延感知自路由规则施加偏置，从而间接影响后续窗口中的流量分配。在这种弱可观测和延迟反馈条件下，本文关注的并非理想化的全局平均性能最优，而是时延敏感型任务风险在各域之间的分布形态。其原因在于，联盟级平均超时率对局部过载并不敏感，即使整体指标看似合理，任务超时仍可能持续集中在少数域内。因此，本文采用基于窗口摘要定义的局部长尾过载严重程度和最差域风险暴露水平来刻画这种结构性风险，同时要求尽力而为型任务的性能不发生明显恶化。基于此，联盟级控制问题被形式化为一个弱可观测条件下的闭环反馈问题，而不是一个拥有全局视野的理想最优化问题。具体地，在每个窗口t结束时，FO 获取观测数据$\{\boldsymbol{s}_j^{t}\}_{j\in\mathcal{D}}$并按如下方式更新通行代价：

$$\tau_{j,s}^{t+1}=F_s\left(\tau_{j,s}^{t},\boldsymbol{s}_j^{t}\right),\ j\in\mathcal{D},s\in\mathcal{S}_{\mathrm{ctl}}。\tag{6}$$

该更新规则仅依赖粗粒度的窗口摘要。其设计要求是确保代价信号有界且平滑变化，从而与基于滞后快照的网关自路由机制相兼容。在下一节中，本文将把F_s具体实例化为PDSF机制，即一种仅由窗口级任务到达率和超时统计驱动的轻量级反馈调节器。

4 基于域级SLO反馈的控制机制

PDSF是一种基于域级SLO反馈的闭环调节机制，其核心思想是在仅具备窗口级摘要的条件下，为各域动态维护有界的路由代价。该代价由FO在窗口边界更新，并广播至各域网关，与滞后的长尾时延快照共同构成最终的路由开销。PDSF的目标是在不侵入域内调度器的前提下，利用这种低维且存在时滞的反馈信号，抑制时延敏感型任务风险在少数域内的持续聚集。

具体而言，当某个域针对时延敏感型任务持续表现出较高的超时压力时，其路由代价会相应增加，从而提高该域在DG视角下的等效路由开销，并降低其在后续窗口中对新任务的吸引力。相反，当该域的超时压力缓解时，其路由代价又会逐步回落，使流量分配重新趋于平衡。整个过程仅依赖窗口级统计摘要，无需感知域内细粒度队列状态，也不要求修改本地调度策略，因此能够与联盟环境下的自治约束相兼容。

4.1 更新规则

在每个控制窗口结束时，各域网关报告窗口级到达量与超时数。设$\ell_{\mathrm{ctl}}\in\{0,1,2,\cdots\}$为以窗口数度量的反馈延迟，在控制时刻$t$，FO将基于最新可获取的观测摘要$\left(A_{j,s}^{t-\ell_{\mathrm{ctl}}},V_{j,s}^{t-\ell_{\mathrm{ctl}}}\right)$来计算下一时刻的路由代价$\tau^{(t+1)}$。该更新针对每个域$d_j$及每个受控类别$s\in\mathcal{S}_{\mathrm{ctl}}$独立执行。

当系统未观测到新的超时压力时，既有的路由代价应逐渐向基准线回落。具体而言，对于每一对(域j，任务类型s)，首先定义衰减后的中间状态为

$$\tilde{\tau}_{j,s}^{t}=clip\left((1-\gamma)\tau_{j,s}^{t},\tau_{\min},\tau_{\max}\right),\tag{7}$$

其中$\gamma\in(0,1)$为全局泄漏因子，$clip(\cdot)$表示显式限幅算子。这种泄漏机制用于避免历史过载状态在路由代价中长期累积。

为便于表述，设$A=A_{j,s}^{t-\ell_{\mathrm{ctl}}}$和$V=V_{j,s}^{t-\ell_{\mathrm{ctl}}}$分别表示最近可观测窗口中的任务到达量与实际超时数。给定类别s的目标超时率θ_s，则该窗口内可容忍的最大超时数为θ_sA。PDSF以超出该预算的部分实际超时数来刻画过载压力，并进一步用期望计数对其进行归一化：

$$E_{j,s}^{t}=\frac{\max(V-\theta_sA,0)}{\max(\theta_sA,1)}。\tag{8}$$

这里，分子中的取正操作$\max(\cdot,0)$保证只有当实际超时数超过目标预算时，才会触发加性惩罚；反之，若$V\leqslant\theta_sA$，则加性修正项归零，此时路由代价仅受式(7)中的泄漏机制主导而缓慢下降。

考虑到突发流量场景下窗口级超额量可能具有明显的长尾特征，PDSF进一步引入有界的非线性压缩函数(squashing function)对惩罚信号进行平滑：

$$e_{j,s}^{t}=\tanh\left(\frac{E_{j,s}^{t}}{\sigma_s}\right),\tag{9}$$

其中$\sigma_s>0$为类特定的缩放参数。由于$E_{j,s}^{t}\geqslant0$，压缩后的信号满足$0\leqslant e_{j,s}^{t}\leqslant1$。此外，为避免低采样窗口引入过大的统计抖动，系统还引入样本量加权机制，以降低低流量窗口对更新的影响：

$$w_{j,s}^{t}=\frac{A}{A+a_0},\tag{10}$$

其中$a_0>0$为全局样本量参数。

综合泄漏衰减、有界超额惩罚和样本量加权3部分，下一时刻的路由代价更新为

$$\tau_{j,s}^{t+1}=clip(\tilde{\tau}_{j,s}^{t}+\eta_sw_{j,s}^{t}e_{j,s}^{t},\tau_{\min},\tau_{\max}),\tag{11}$$

其中$\eta_s>0$为控制通行代价对超额超时信号的响应幅度；较大的η_s会带来更激进的更新，而较小的η_s则对应更保守的调节。上述FO端单次代价更新的完整过程可由算法1概括。

算法1. PDSF更新函数。

输入：窗口索引t及反馈滞后ℓ_{ctl}，域集合$\mathcal{D}$及受控类别集合$\mathcal{S}_{\mathrm{ctl}}$，当前通行代价$\{\tau_{j,s}^{t}\}_{j\in\mathcal{D},s\in\mathcal{S}_{\mathrm{ctl}}}$，最新统计摘要$\{A_{j,s}^{t-\ell_{\mathrm{ctl}}},V_{j,s}^{t-\ell_{\mathrm{ctl}}}\}_{j\in\mathcal{D},s\in\mathcal{S}_{\mathrm{ctl}}}$，参数$\{\theta_s,\eta_s,\sigma_s\}_{s\in\mathcal{S}_{\mathrm{ctl}}}$以及$\gamma,a_0,\tau_{\min},\tau_{\max}$；

输出：$\{\tau_{j,s}^{t+1}\}_{j\in\mathcal{D},s\in\mathcal{S}_{\mathrm{ctl}}}$。

① for $j\in\mathcal{D},s\in\mathcal{S}_{\mathrm{ctl}}$ do

② $\tilde{\tau}\leftarrow clip\left((1-\gamma)\tau_{j,s}^{t},\tau_{\min},\tau_{\max}\right)$;

③ $A\leftarrow A_{j,s}^{t-\ell_{\mathrm{ctl}}}$, $V\leftarrow V_{j,s}^{t-\ell_{\mathrm{ctl}}}$;

④ $T\leftarrow\max(\theta_s\cdot A,1)$;

⑤ $E\leftarrow\max(V-\theta_s\cdot A,0)/T$;

⑥ $e\leftarrow\tanh(E/\sigma_s)$;

⑦ $w\leftarrow A/(A+a_0)$;

⑧ $\tau_{j,s}^{t+1}\leftarrow clip(\tilde{\tau}+\eta_s\cdot w\cdot e,\tau_{\min},\tau_{\max})$;

⑨ end for

⑩ return $\{\tau_{j,s}^{t+1}\}_{j\in\mathcal{D},s\in\mathcal{S}_{\mathrm{ctl}}}$。

4.2 性质分析

式(7)及式(11)中的更新旨在满足联盟环境下“低维、缓变、有界”的控制要求，并在仅具备延迟窗口摘要的条件下构建一个可解释的反馈通道。首先，限幅算子将通行代价显式约束在区间$[\tau_{\min},\tau_{\max}]$，从而避免其在持续的过载压力下无界增长。其次，乘法因子$1-\gamma$引入了泄漏机制。当某个域在给定窗口内的实际超时数未超过目标预算时，式(8)中的超额量$E^t_{j,s}$取零，加性惩罚项随之消失，此时路由代价将以几何级数逐渐衰减至基准线。因此，该机制既能够对持续存在的过载压力保持一定记忆，又能够在压力消退后逐渐遗忘瞬时峰值。进一步地，PDSF并不直接使用绝对超时数作为驱动信号，而是采用$\max(\theta_s A,1)$对超额量进行归一化。这一处理使不同任务规模、不同负载水平下的更新幅度具备可比性，从而避免高到达量域天然获得更强惩罚。此外，样本量加权因子$w^t_{j,s}=A/(A+a_0)$会降低低采样窗口对更新的影响，因此能够减轻噪声窗口和冷启动域带来的统计抖动。换言之，PDSF所施加的并不是对瞬时异常的激进响应，而是一种经过归一化和加权后的平滑负反馈。最后，PDSF并不依赖于某一种特定的入口路由算法，而只要求受控类别满足基本的单调性条件，即在其他条件保持不变时，单方面提高目标域的通行代价$\tau_{j,s}$，不会导致该域获得更多同类流量。正如第3.4节所述，Softmin路由和确定性的最小代价路由都满足这一条件。因此，通行代价可以在不暴露域内排队状态和本地调度细节的前提下稳定地发挥负反馈信号的作用。

4.3 理论性质

PDSF的定位是在窗口化且存在测量延迟条件下的实用型闭环调节器。基于标准流体近似的局部分析，有助于从理论层面解释该机制为何能在弱单调性假设下维持良好的动态行为。沿用排队系统反馈控制的经典分析范式[12,25]，本文通过条件期望研究窗口级反馈，并将其他域及尽力而为型任务的影响视为环境扰动。在控制时刻t，系统采用有界的反馈信号$q^t=w^t e^t$，其中$w^t\in[0,1]$为式(10)中的样本量权重，$e^t\in[0,1]$为式(9)经非线性压缩后的超额量。由于双曲正切函数$\tanh(\cdot)$的性质，且式(8)中的超额量在构造上严格非负，反馈信号必然满足$q^t\in[0,1]$。为聚焦于系统的核心动态，本文暂时忽略硬性限幅操作，将单维更新规则写为

$$\tau^{t+1}=(1-\gamma)\tau^t+\eta q^t, \tag{12}$$

其中$\eta>0$为受控类别的步长。此处的延迟依赖性源于q^t是基于最新可获取的窗口摘要计算而来，而该摘要又由更早时刻路由决策时所诱导的系统状态决定。

引理 1. 有界输入响应。

对于式(12)中的未限幅更新，路由代价始终满足：

$$\forall t,0\leqslant\tau^t\leqslant\max\left(\tau^0,\frac{\eta}{\gamma}\right)。 \tag{13}$$

若进一步叠加显式限幅操作，则对于任意时间步t，均有$\tau^t\in[\tau_{\min},\tau_{\max}]$。

证明. 因为反馈信号$q^t\in[0,1]$，根据式(12)可得$\tau^{t+1}\leqslant(1-\gamma)\tau^t+\eta$且显然有$\tau^{t+1}\geqslant 0$。令$B=\max\{\tau^0,\eta/\gamma\}$。若在时刻$t$有$\tau^t\leqslant B$，则$(1-\gamma)\tau^t+\eta\leqslant(1-\gamma)B+\eta\leqslant B$。由数学归纳法可知，对所有的$t$均有$\tau^t\leqslant B$。若考虑限幅，则绝对边界由式(6)和式(11)中$clip(\cdot)$定义给出。 证毕.

引理 2. 无过载状态下的系统遗忘性。

假设在未限幅更新下，对连续时间步$k=t,t+1,\cdots,t+K-1$均有$q^k=0$，则$\tau^{t+K}=(1-\gamma)^K\tau^t$。若考虑限幅操作，这种基于几何衰减形式的均值回归特性依然成立，直至变量被投影至边界区间$[\tau_{\min},\tau_{\max}]$内。

证明. 当$q^k=0$时，式(12)直接退化为一阶齐次线性递推关系$\tau^{k+1}=(1-\gamma)\tau^k$。通过对该递推式连续展开，即可得到上述指数衰减表达式。

为了刻画系统在延迟反馈环境下的局部动态，本文进一步用条件期望响应随机观测信号。设$\ell_{\mathrm{ctl}}\geqslant 0$表示以窗口数量为度量单位的反馈滞后周期，定义期望响应函数$h(\tau)=\mathbb{E}[q^t\mid\tau^{t-\ell_{\mathrm{ctl}}}=\tau]$。本文的核心建模假设在于，假设是$h(\tau)$在系统平衡工作点附近局部连续可微，且$\tau$为非增函数。这一假设既与路由对通行代价的单调响应一致，也符合τ作为超时超额惩罚信号的物理意义。在忽略硬限幅的前提下，期望状态的更新动态可写为：

$$\tau^{t+1}=(1-\gamma)\tau^t+\eta h\left(\tau^{t-\ell_{\mathrm{ctl}}}\right)。 \tag{14}$$

证毕.

命题 1. 延迟反馈下的局部渐进稳定性。

假设期望响应函数$h(\tau)$在式(14)的不动点τ^*的邻域内连续可微且非增。定义$a=1-\gamma$和$b=\eta h'(\tau^*)$。若满足条件

$$|a|+|b|<1, \tag{15}$$

则式(14)在不动点τ^*附近的线性化系统对于任意固定的延迟常数ℓ_{ctl}都是渐进稳定的。

证明. 引入误差项$e^t=\tau^t-\tau^*$。在τ^*处对h进行一

阶泰勒展开，可得$h(\tau^{t-\ell_{\mathrm{ctl}}}) \approx h(\tau^*)+h'(\tau^*)e^{t-\ell_{\mathrm{ctl}}}$。由于$\tau^*$是系统的不动点，展开后的常数项相互抵消，从而线性化误差动态$e^{t+1} \approx ae^t + be^{t-\ell_{\mathrm{ctl}}}$。

进一步定义时间滑动窗口内的最大绝对误差$M^t = \max\{|e^t|,|e^{t-1}|,\cdots,|e^{t-\ell_{\mathrm{ctl}}}|\}$。利用三角不等式有$|e^{(t+1)}| \leqslant |a||e^t|+|b||e^{t-\ell_{\mathrm{ctl}}}| \leqslant (|a|+|b|)M^t$。

在式(15)条件下，令$c=|a|+|b|$，显然有$c<1$。由M^t的定义可知序列满足$M^{t+1} \leqslant M^t$，且$|e^{t+1}| \leqslant cM^t$。经过$\ell_{\mathrm{ctl}}+1$个离散步长后，滑动窗口中的最大值集合中将不再包含初始元素e^t，进而有

$$M^{t+\ell_{\mathrm{ctl}}+1} = \max\left\{|e^{t-\ell_{\mathrm{ctl}}+1}|,\cdots,|e^{t+1}|\right\} \leqslant cM^t \text{。}$$

这意味着，包络序列M^t在沿子序列$t, t+\ell_{\mathrm{ctl}}+1, t+2(\ell_{\mathrm{ctl}}+1),\cdots$呈几何衰减，故线性化系统在不动点领域内渐进稳定。 证毕.

综上所述，引理1和引理2在不显式建模底层排队过程的前提下，分别刻画了PDSF的有界性和遗忘性；命题1则给出了反馈延迟条件下的局部稳定性充分条件，并揭示了有效控制增益与泄漏机制之间的耦合关系。由于$h'(\tau^*) \leqslant 0$，式(15)在物理上等价于$\eta|h'(\tau^*)| < \gamma$，这意味着在实践中不宜采用过大的控制增益而需同时配置过弱的泄漏，否则可能削弱延迟反馈下控制环路的稳定性。η_s的选择需要与泄漏强度γ协同考虑，以避免延迟反馈下控制环路过于敏感。需要强调的是，这一稳定性条件是局部的，并不意味着系统在极端非平稳扰动下仍具备全局收敛性；它说明的是，在底层入口路由仅满足弱单调响应的前提下，PDSF依然能够通过有界、缓变的方式维持联盟级控制环路的稳定演化。

5 实验结果

5.1 轨迹驱动的工作负载

本文的实验采用基于Google集群轨迹构建的半合成数据集[15]。为了评估窗口级控制机制，实验将连续提交时间离散映射为时间分片(time bins)，从而形成标准化的窗口数据流，同时尽可能保留原始到达过程的非平稳性与短时突发特征。图4给出了各时间分片内的任务到达量，可以看到在缓慢变化的整体趋势之上有明显的短时峰值。原始公开轨迹对应于8个逻辑域，实验中进一步将其重构为5个自治域，以形成一个异构程度可控且域数量适中的评估环境。

鉴于原始轨迹在部分遥测字段上的缺失，为了支持多域环境下的时延与开销建模，本文进一步为每个任务补充了一组基础语义属性，包括数据大小、I/O强度以及可并行化参数。具体而言，任务首先根据经验性的CPU-内存比例分布，被划分为CPU密集型、均衡型或内存密集型3类。随后从经校准的类型分布中抽取附加属性，以复现真实资源混合特征。最后，本文引入类似阿姆达尔定律的计算模型，在考虑并行化收益递减及数据I/O约束的基础上生成理想处理时间。这些细粒度语义属性仅供底层黑盒域执行模型使用，FO对其不可见，联盟层只能访问窗口级统计摘要。后续所有核心实验均在上述5个自治域架构上展开，除专门的敏感性分析外，其默认设定汇总于表3。

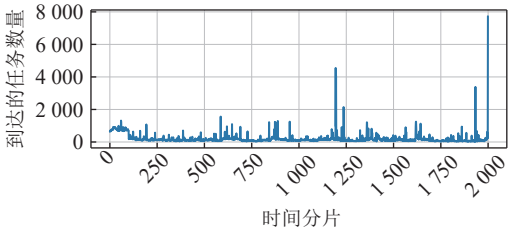


Fig. 4 Task arrivals of each time bin in the trace-driven semi-synthetic workload

图4 轨迹驱动半合成负载中各时间分片的任务到达量

Table 3 Primary Experimental Settings

表3 主实验设定

项目	设定
自治域数量	5
时间分片数量	2 000 时间片
主评估区间长度	1 500 时间片
控制窗口长度	每个窗口3时间片
快照滞后ℓ	2
反馈延迟ℓ_{ctl}	0
随机种子	{7,8,9,10,11}
步长参数η_{LC}，η_{BE}	50，10
泄漏因子γ	0.15
样本量参数a_0	150
通行代价边界$[\tau_{\min},\tau_{\max}]$	[0,2 000]

5.2 基线与参考策略

实验将PDSF与多种可部署的基准方法，以及2种具备全局视野的理想参考策略进行对比。所有可部署方法均采用相同的联盟接口与路由规则，且仅依赖联盟层可见的窗口摘要与滞后的时延快照。它们的差异主要体现在2方面，即是否对快照进行平滑处理，以及是否生成域级路由代价。具体方法如下。

1）无控制（no control, NC）。固定$\tau \equiv 0$, 仅保留基于快照的自路由行为, 不施加任何联盟级的反馈干预。

2）缺口通行代价 SLA（shortfall toll）。仅基于最近一个窗口的超时缺口, 施加无记忆的惩罚。设定$\hat{v}_{j,\mathrm{LC}} = V_{j,\mathrm{LC}}/(A_{j,\mathrm{LC}} + a_0)$则更新规则为$\tau_{j,\mathrm{LC}} = \alpha \max(\hat{v}_{j,\mathrm{LC}} - \theta_{\mathrm{LC}}, 0)$。

3）过滤快照（EWMA）。保持$\tau \equiv 0$, 但采用指数加权移动平均对滞后快照进行平滑以考察单纯过滤快照噪声所带来的收益[40]。

4）AIMD 通行代价。当窗口超时率超过目标阈值时, 加法增大$\tau_{j,\mathrm{LC}}$; 否则乘法减小$\tau_{j,\mathrm{LC}}$。该策略沿用了经典的拥塞控制与分布式自适应思想[40]。

5）PDSF。采用本文提出的有界泄漏积分更新机制, 由窗口级到达量与超时统计共同驱动域级路由代价。

6）时延敏感全知策略（latency-critical oracle, LC-O）。利用全局离线信息计算固定路由分布, 以最小化时延敏感型任务风险。该策略突破了快照滞后与自路由限制, 但可能显著牺牲 BE 任务性能。

7）权衡全知策略 TO-O（tradeoff oracle）。同样基于离线全局优化, 但在 LC 任务与 BE 任务的性能之间寻求权衡。它仅作为系统性能的理论上限基准, 而非实际可部署策略。

5.3 结构性风险度量指标

在总体服务指标方面, 本文采用联盟级平均超时率进行评估。然而, 延迟的窗口反馈容易诱发局部过载聚集, 使少数恶劣窗口主导系统风险, 因此仅依赖平均指标往往难以反映弱可观测条件下真实的风险形态。为更直观地说明这一点, 下面给出一个简单例子。设某一窗口内, 5 个域各接收 200 个 LC 任务, 因此联盟 LC 任务总到达量为 1 000。若类级目标超时率为θ_{LC}=0.06, 则每个域在该窗口内可容忍的截止期违例预算为 0.06×200=12 个。现在考虑 2 个系统, 它们的联盟平均超时率均为 5%, 即总截止期违例数均为 50。系统 *A* 中, 5 个域的截止期违例数分别为 10, 11, 9, 12, 8; 系统 *B* 中, 5 个域的截止期违例数分别为 0, 0, 0, 5, 45。显然, 两者的平均超时率完全相同, 但其系统含义截然不同。在系统 *A* 中, 违例在各域之间较为均匀分散; 而在系统 *B* 中, 第 5 个域单独承担了 45 个违例, 超出其预算 33 个, 对应的归一化超额违例比率达到 (45−12)/12=2.75。这表明, 平均超时率相同并不意味着系统风险相同, 系统 *B* 反映的是更严重的局部拥塞、更高的最差域暴露以及更强的瞬态失稳风险。基于这一观察, 本文进一步引入局部长尾过载严重程度与最差域风险暴露水平 2 个指标, 以刻画平均指标无法揭示的结构性风险。设$A_{j,\mathrm{LC}}^{t}$和$V_{j,\mathrm{LC}}^{t}$表示在窗口t内路由至域d_j的 LC 任务到达量与实际超时数。在整个评估周期内, 联盟级的宏观超时率定义为

$$\bar{v}_{\mathrm{LC}} = \frac{\sum_{t}\sum_{j\in\mathcal{D}} V_{j,\mathrm{LC}}^{t}}{\sum_{t}\sum_{j\in\mathcal{D}} A_{j,\mathrm{LC}}^{t} + \varepsilon},$$

其中$\varepsilon > 0$用于防止除零异常。

1）长尾汇聚严重程度

该指标用于衡量单个窗口内最差域超出其 LC 任务超时预算的程度, 从而刻画间歇性的局部过载突发。对于窗口t中的域d_j, 首先定义其超出目标预算$\theta_{\mathrm{LC}}A_{j,\mathrm{LC}}^{t}$的绝对超额量为$E_j^t = \max\left(V_{j,\mathrm{LC}}^{t} - \theta_{\mathrm{LC}}A_{j,\mathrm{LC}}^{t}, 0\right)$, 为了消除负载规模带来的影响, 用目标违约数对其进行归一化, 得到$R_j^t = \dfrac{E_j^t}{\max\left(\theta_{\mathrm{LC}}A_{j,\mathrm{LC}}^{t}, 1\right)}$。同时, 为避免极小样本带来的统计放大效应, 引入最小到达量过滤器, 定义有效域集合$\mathcal{D}_t = \{j \in \mathcal{D} \mid A_{j,\mathrm{LC}}^{t} \geqslant A_{\min}\}$。则窗口$t$的长尾汇聚严重程度定义为$S^t = \max_{j\in\mathcal{D}_t} R_j^t$, 在统计$S^t$的长尾百分位时, 仅保留具备足够总风险暴露的窗口（即$\sum_j A_{j,\mathrm{LC}}^{t} \geqslant A_{\min}$）集合。

2）最差域风险暴露水平

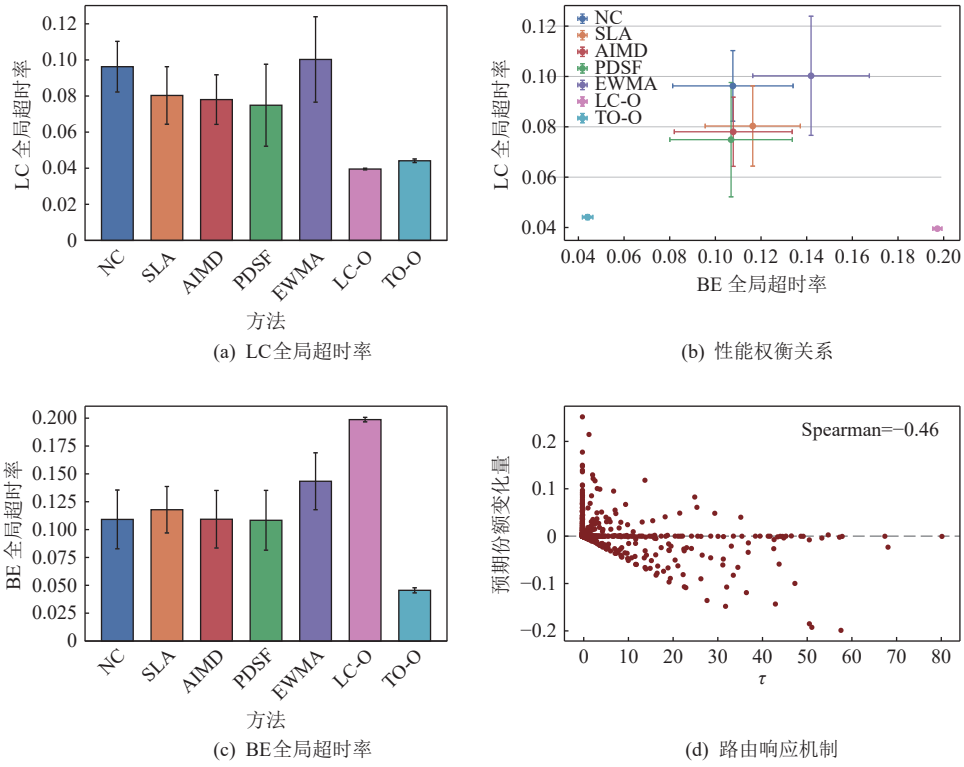
该指标旨在刻画贯穿整个评估周期的持久性风险集中现象, 是对窗口级长尾汇聚严重程度的重要补充。设$A_{j,\mathrm{LC}} = \sum_t A_{j,\mathrm{LC}}^{t}$和$V_{j,\mathrm{LC}} = \sum_t V_{j,\mathrm{LC}}^{t}$分别表示域$d_j$在全局评估周期上的聚合到达量与总超时数。进一步定义有效聚合域集合$\mathcal{D}_{\mathrm{agg}} = \{j \in \mathcal{D} \mid A_{j,\mathrm{LC}} \geqslant A_{\min}\}$, 则最差域风险暴露水平计算$W = \max\limits_{j\in\mathcal{D}_{\mathrm{agg}}} \dfrac{V_{j,\mathrm{LC}}}{A_{j,\mathrm{LC}}}$。

5.4 实验结果

实验在 2 个时间尺度上展开。首先, 在轨迹预处理阶段, 任务到达过程被离散为细粒度时间分片。其次, 在联盟层的遥测与控制阶段, 系统运行在由多个时间分片聚合而成的粗粒度时间窗口上。在每个窗口结束时, 域网关向联盟层上报汇总统计信息。若引入ℓ个窗口的快照滞后, 则窗口t内的路由决策实际上使用的是来自窗口$t-\ell$的快照输入。为了避免在评估时人为挑选有利于特定算法的延迟参数, 本文以一种与具体方法无关的方式, 直接从 NC 基准的运行结果中确定主实验所采用的滞后值, 即

$\ell^* = \arg\max \mathrm{TailRisk}_{\mathrm{NC}}(\ell)$，其中 $\mathrm{TailRisk}_{\mathrm{NC}}(\ell)$实例化为基于$S^{(\ell)}$计算的第 99 百分位长尾汇聚严重程度，本文的主实验结果均在ℓ^*处报告，同时给出关于ℓ的完整参数扫描，以刻画系统的边界状态。在当前的工作负载下，测得$\ell^* = 2$，即主实验采用滞后 2 个窗口的快照。该设定对应跨域状态上报与数据聚合带来的实际遥测延迟，此时最新的底层队列状态对联盟层不可见。

图 5 报告了各方法在联盟级平均超时率上的表现，并给出了路由干预机制的直接证据。图 5(a)表明，与 NC 相比，PDSF 能显著降低 LC 任务的平均超时率，同时并未导致 BE 任务的超时率明显上升，这说明其性能增益并非通过转移或牺牲 BE 任务获得。相比之下，EWMA 仅对滞后快照进行平滑，而不引入额外的路由代价，因此在当前设定下难以形成与 PDSF 可比的工作点。2 种具备全局信息的理想策略仅作为理论参考，而非可部署方法。其中，LC-O 虽然大幅降低了 LC 任务的超时率，却明显推高了 BE 任务的超时风险，说明单纯追求单一类别指标最优会带来显著副作用；TO-O 则通过离线全局优化在 LC 与 BE 之间实现了更均衡的折中，因而可视为一种理论上的性能上界。



(a) LC全局超时率　(b) 性能权衡关系

(c) BE全局超时率　(d) 路由响应机制

Fig. 5 Primary outcomes and mechanistic evidence with window-level observability

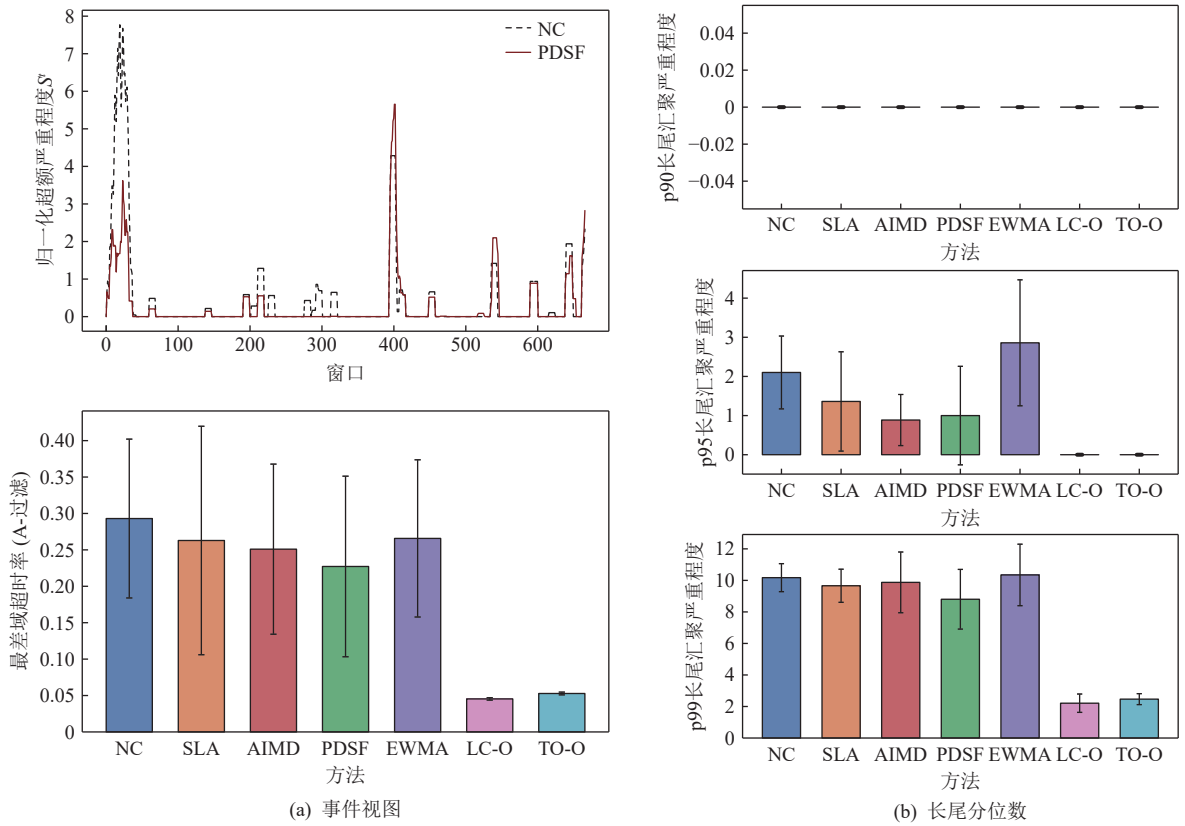
图 5 窗口级可观测性下的主要结果与机制证据

图 5(b)进一步揭示了 PDSF 的作用通道。上半部分比较各方法对 LC 与 BE 的联盟级超时率的影响，直观展示了它们隐含的性能权衡关系。下半部分则考察路由代价如何改变路由偏好。考虑到快照波动和路由本身的随机性会干扰对观测份额的直接比较，本文采用反事实方式进行评估，即在保持输入任务和快照完全一致的条件下，将控制组的路由代价强制设为 0，并比较由此产生的预期路由份额变化。结果表明$\tau^{t}_{j,\mathrm{LC}}$与反事实份额变化之间存在稳定的负相关关系，这与第 3.4 节中的单调性假设一致。即 PDSF 是通过提高局部拥塞域的等效路由开销，促使自路由流量在明显过载出现之前提前发生偏移。传统的平均指标容易掩盖长尾风险，而结构性风险指标则能有效将典型窗口与那些主导了严格超时风险的罕见过载事件区分开来。以下视图均聚焦于 LC 任务，同时引入了最小到达量过滤器$A_{\min}$以防止分母过小引发的统计畸变。

传统平均指标往往会掩盖长尾风险，而结构性风险指标能够将普通窗口与主导严格超时风险的少数极端过载事件区分开来。以下分析均聚焦于 LC 任务，并沿用第 5.3 节中的指标定义，同时采用最小到达量过滤阈值$A_{\min}$，以避免样本量过小造成统计失真。

图 6(a)的事件时序图表明，超额超时现象并非均匀分布，而是高度集中在少数恶劣窗口中。上方子图给出了窗口级长尾汇聚严重程度S^t，即窗口t内各域归一化超额量的最大值。可以看到，大多数窗口的超额量接近于 0，而少数窗口则形成了尖锐的风险峰值。引入 PDSF 后，这些最严重的过载峰值被明显压低，同时高严重度区间的持续时间也显著缩短。下方子图则给出了在流量充足域上计算得到的最差域超时率，用以观察过载是否持续锁定在同一高风险域上。结果表明，PDSF 不仅削弱了瞬时峰值，也降低了风险长期集中在单一域上的程度。图 6(b)给出了S^t的跨窗口分布。从低分位到中分位区间，各方法之间差异并不显著；但在高分位，性能差距迅速拉大。也就是说，各方法在“典型窗口”中的表现相近，而真正拉开差距的是少数极端过载事件。综合图 6(a)与图 6(b)可见，PDSF 的优势并不主要体现在进一步压低日常平均超时率，而在于显著缓和了极端过载窗口中的风险聚集，并有效降低了最差域的风险暴露。

图 7 进一步展示了超时责任在各域之间的分配情况。需要说明的是，由于图中每一列都经过了归一化处理，因此集中分布反映的是剩余超时任务在少数域上的风险聚集，而不等同于原始流量份额本身的不均衡。可以看到，在 NC 基线下，超时高度集中于少数几个域，这说明仅依赖滞后快照的自路由机制容易在联盟局部反复放大负载压力。相比之下，PDSF 显著降低了高风险窗口中持续受压域所承担的超时比例，从而有效打破了这种病态的聚集形态。需要强调的是，这种“打破聚集”并不意味着追求绝对均匀的流量分配。由于底层域在处理能力和运行条件上本就存在异构性，即使是 LC-O 和 TO-O 这类理想策略，其流量份额也可能表现出明显集中。



(a) 事件视图 (b) 长尾分位数

Fig. 6 Structural risk perspectives for delay-sensitive tasks

图 6 时延敏感型任务的结构性风险视角

为在真实轨迹驱动评估之外进一步检验算法的鲁棒性，本文还设计了 2 个极限微基准。其一，阶跃负载基准通过突然注入额外负载制造短时冲击，用以隔离并评估算法的瞬态响应能力。该基准重点考察 LC 任务在最坏情形下的风险峰值，以及压制这些峰值所需的最大控制代价。其二，稳态近饱和基准在剔除预热阶段后，使系统长期维持在高负载压力下，用于刻画持久压力下的长期行为。该基准重点关注长尾汇聚严重程度，以及路由代价信号的波动幅度和变异性。窗口级联盟范围的 LC 超时率定义为

$$\bar{v}_{\mathrm{LC}}=\frac{\sum_{j\in\mathcal{D}}V_{j,\mathrm{LC}}^{t}}{\sum_{j\in\mathcal{D}}A_{j,\mathrm{LC}}^{t}+\varepsilon}。$$

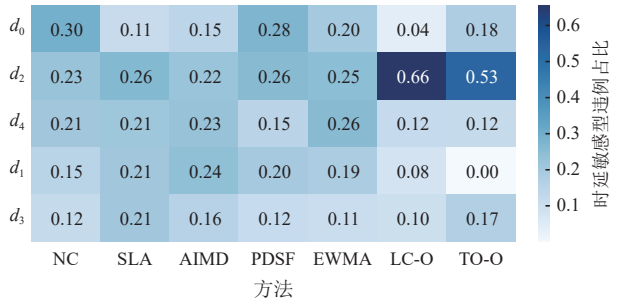


Fig. 7 Distribution of deadline misses of delay-sensitive tasks across domains

图 7 时延敏感型任务截止期违例在各域间的分布

表 4 的阶跃负载基准表明, PDSF 在有效抑制最坏情况过载风险的同时, 其引发的路由代价峰值明显低于 AIMD 策略。具体而言, 与 AIMD 相比, PDSF 降低了 p99($\hat{v}_{\mathrm{LC}}^{t}$)、峰值$\max_t\hat{v}_{\mathrm{LC}}^{t}$以及峰值事件严重程度$\max_t S^t$, 同时显著压低了峰值控制代价$\max_t\tau_{\max}^{t}$, 这表明系统无需依赖剧烈的代价尖峰, 即可有效缓解突发的过载事件。在该基准测试下, 2 种控制器的稳定恢复时间基本相当。此外, 为验证代价更新中记忆衰减项的必要性, 微基准中还引入了 PDSF-NL 变体, 即强制关闭泄漏机制, 作为一种“长记忆”极端对照。

Table 4 Step-Load Micro-Benchmark Test

表 4 阶跃负载微基准测试

方法	p99($\hat{v}_{\mathrm{LC}}^{t}$)	$\max_t\hat{v}_{\mathrm{LC}}^{t}$	$\max_t S^t$	$\max_t\tau_{\max}^{t}$
AIMD	0.857 8	0.975 5	15.258 0	223.617 6
PDSF	0.807 9	0.944 4	14.740 4	134.469 6
PDSF-NL	0.794 6	0.976 4	15.273 2	568.878 0

注: S^t表示窗口级长尾汇聚严重程度。控制代价则以窗口内最大的 LC 任务通行代价概括, 即$\tau_{\max}^{t}=\max_{j\in\mathcal{D}}\tau_{j,\mathrm{LC}}^{t}$, 其离散变化率定义为$\Delta\tau_{\max}^{t}=\tau_{\max}^{t}-\tau_{\max}^{t-1}$。

表 5 的稳态近饱和微基准刻画了系统在持续高压下的长期表现。结果显示, PDSF 在典型状态和高分位数状态下均有效降低了 LC 任务的风险, 并生成了更平滑的路由代价轨迹。p99($\hat{v}_{\mathrm{LC}}^{t}$) 与 $\mathrm{std}_t(\Delta\tau_{\max}^{t})$的同步下降表明, PDSF 在持续高负载环境中能够避免代价信号出现大幅振荡。相比之下, PDSF-NL 虽然能够进一步压低高风险状态下的长尾指标, 却明显放大了控制动作的幅度和波动性。这一结果从侧面说明, 在主实验中保留默认泄漏配置是必要且合理的。

为了进一步探究算法的鲁棒性与边界条件, 本

Table 5 Steady Near-Saturation Micro-Benchmark Test

表 5 稳态近饱和微基准测试

方法	p95($\hat{v}_{\mathrm{LC}}^{t}$)	p99($\hat{v}_{\mathrm{LC}}^{t}$)	p95(S^t)	p99($\tau_{\max}^{t}$)	$\mathrm{std}_t(\Delta\tau_{\max}^{t})$
AIMD	0.20	0.59	2.561 2	238.25	13.50
PDSF	0.07	0.54	0.93	54.47	6.50
PDSF-NL	0.05	0.51	0.17	576.81	19.40

注: S^t表示窗口级长尾汇聚严重程度。控制代价则以窗口内最大的 LC 任务通行代价概括, 即$\tau_{\max}^{t}=\max_{j\in\mathcal{D}}\tau_{j,\mathrm{LC}}^{t}$, 其离散变化率定义为$\Delta\tau_{\max}^{t}=\tau_{\max}^{t}-\tau_{\max}^{t-1}$。

文从联盟路由接口的 2 个核心维度展开了敏感性分析。第 1 个维度是网关处的状态快照滞后程度(第 3.4 节中定义的快照滞后ℓ)。第 2 个维度是 Softmin 路由下的路由集中度, 其大小由 LC 任务的逆温度参数β_{LC}进行调控。在这 2 组参数扫描实验中, 监控窗口的长度与控制周期保持恒定, 最终结果在多个随机种子上取平均。

在第 1 组实验中, 系统通过对$\ell\in\{1,2,3,4\}$进行扫描来改变快照陈旧度。结果表明, 系统对延迟的敏感性呈现出非单调特征。中等程度的滞后会使 NC 基线在尾部严重程度上出现最明显的退化, 而在这一范围内, PDSF 能显著降低长尾汇聚严重程度, 并表现出与 AIMD 相当的竞争力。当滞后进一步增大时, 各方法之间的差距开始收敛, AIMD 的相对竞争力上升, 这表明控制环路对近期负载波动的响应已经显著变钝, 即系统逐渐接近控制边界。在这一阶段, 保留泄漏机制的 PDSF 仍优于 NC, 同时其调节幅度始终明显低于长记忆极端情形。

图 8 给出了上述快照滞后扫描的结果。其中, 图 8(a)展示了长尾汇聚严重程度随ℓ的变化趋势, 图 8(b)则对应实现这些收益所需的控制代价。2 个图共同表明, PDSF 在轻度到中度滞后范围内具有最稳定的收益, 而在严重滞后条件下, 其优势会逐渐收窄。

第 2 组实验通过缩放β_{LC}来改变路由集中度。较大的β_{LC}迫使路由高度集中于成本最低的域, 从而极大地增加了在滞后快照下发生“同步拥塞”的风险。需要指出的是, 这一温度参数扫描的目的在于刻画方法的适用边界, 而非单纯进行参数调优。结果表明, PDSF 在中等集中度下对长尾风险的缓解最为显著。此时, 混合路由仍保留了足够的流量多样性, 使得有界路由代价能够在不触发全局突发切换的前提下平滑地重塑流量份额。然而, 当路由逐渐逼近确定性选择, 即β_{LC}变得很大时, PDSF 的性能优势逐渐收窄, 而达成这些收益所需的调节动作却迅速增大,

这表明基于快照的反馈调节在高度集中路由条件下存在明确的效用边界。

图 9 展示了路由集中度扫描的具体结果。其中，图 9(a)报告长尾汇聚严重程度的变化，图 9(b)给出相应的控制代价。两者共同说明，PDSF 更适合作为中等路由平滑程度下的默认调节机制，而不适合被理解为在任意集中度条件下都保持同等优势的统一方案。

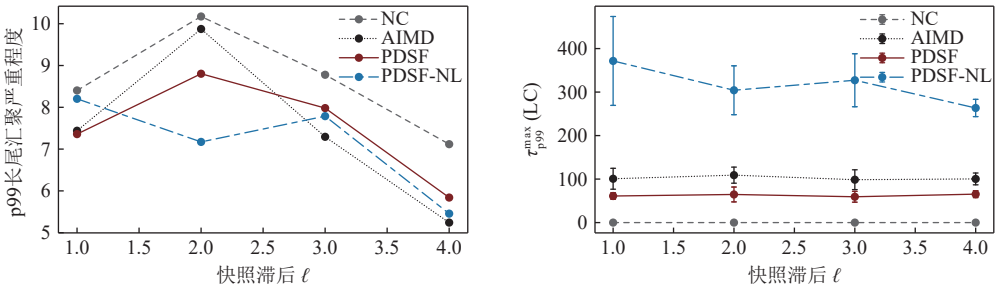


Fig. 8 Sensitivity to snapshot lag ℓ and the corresponding actuation cost

图 8 对快照滞后ℓ的敏感性及相应的控制代价

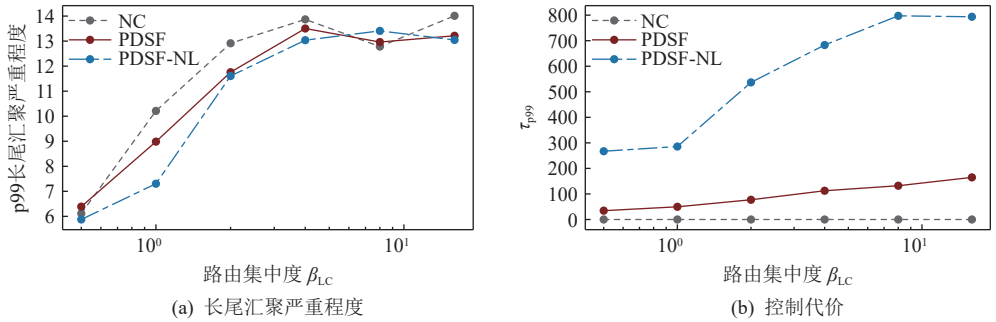


Fig. 9 Sensitivity to routing concentration controlled by β_{LC}

图 9 由β_{LC}控制的路由集中度的敏感性

表 6 对信息的陈旧度状态转变进行了经验性总结。总体而言，这 2 组扫描实验表明，测量滞后性与路由反应性之间的耦合作用会共同决定系统所处的运行区间。在轻度和中度滞后条件下，PDSF 提供了更适合作为默认方案的工作点；延长记忆虽可进一步压制高风险尾部事件，但往往伴随更高的调节代价；而在极度严重的滞后环境下，具备较强抗延迟特性的基准方法，如 AIMD，则会表现出更强的竞争力。

Table 6 Empirical Guidelines Principles Proposed by Observed Regime Shifts

表 6 基于观测到的状态转变提出的经验指导原则

陈旧度状态	典型选择
轻度ℓ	PDSF
中度ℓ	PDSF-NL
重度ℓ	AIMD

5.5 讨 论

从敏感性分析可以看出，PDSF 的收益依赖于信息时效与路由平滑性的共同作用。总体而言，当快照陈旧度处于轻度或中度，且入口路由仍保留一定平滑性时，窗口级摘要能够提供足够的方向性信息，此时有界通行代价可以持续而稳定地重塑流量份额，因此 PDSF 更适合作为联盟级控制的默认方案。相反，当快照进一步陈旧或路由逐渐逼近确定性选择时，低维反馈所能利用的有效信息显著减少，PDSF 的优势也会随之收窄。其根本原因在于，PDSF 通过泄漏、非线性压缩和样本量加权来换取更好的稳定性与可解释性，但这些设计同时也降低了极端滞后条件下的有效控制增益。相比之下，AIMD 保留了更强的动作记忆，因此在严重陈旧场景下可能表现出更强的竞争力。由此可见，本文机制更适用于中等信息时效和中等路由平滑条件下的联盟环境；而当快照极度陈旧、入口路由近乎确定，或域间性能在窗口内快速反转时，其调节空间将明显受限。

6 结 论

本文聚焦于弱可观测条件下的联盟算力网多域

路由问题。在该架构中，联盟控制层只能获得底层自治域周期性上报的窗口级聚合摘要以及滞后的状态快照。受此接口约束，时延敏感型任务的超时风险容易在少数时间窗口和特定自治域内发生结构性聚集。即使联盟整体的平均性能指标看似处于合理水平，这种局部聚集仍可能引发显著的瞬态过载，并抬升最差域的风险暴露。为在仅具备有限状态信息且不侵入底层域内调度器的前提下缓解这一问题，本文提出了域级 SLO 反馈调节机制 PDSF。作为一种基于路由代价的轻量级控制信号，PDSF 仅依赖滞后的窗口摘要进行动态更新，并通过等效调整网关视角下的路由开销，平滑引导各网关的自路由决策。基于真实集群工作负载轨迹的实验结果表明，在相同可观测性约束下，PDSF 能够显著缓解长尾过载的严重程度，并在保护时延敏感型任务的同时避免尽力而为型任务性能出现明显退化。

作者贡献声明：周钰雯负责整体实验代码并撰写论文；任棒棒、何瑞负责实验方案设计和论文修改；郭得科提出整体指导意见和算法思路。

参 考 文 献

[1] Sun Yukun, Lei Bo, Liu Junlin, et al. Computing power network: A survey[J]. China Communications, 2024, 21(9): 109−145

[2] ITU-T SG13. Y. 2501: Computing power network – framework and architecture for consent[R]. Geneva: International Telecommunication Union, 2021

[3] Xie Renchao, Feng Li, Tang Qinqin, et al. Delay-prioritized and reliable task scheduling with long-term load balancing in computing power networks[J]. IEEE Transactions on Services Computing, 2024, 17(6): 3359−3372

[4] Stoica I, Shenker S. From cloud computing to sky computing[C]// Proc of the Workshop on Hot Topics in Operating Systems. New York: ACM, 2021: 26−32

[5] Bernstein D, Ludvigson E, Sankar K, et al. Blueprint for the intercloud-protocols and formats for cloud computing interoperability[C]//Proc of the 4th Int Conf on Internet and Web Applications and Services. Piscataway, NJ: IEEE, 2009: 328−336

[6] Gaia-X European Association for Data and Cloud AISBL. Gaia-X architecture document-22.04 release[R]. Brussels: Gaia-X European Association for Data and Cloud AISBL, 2022

[7] Curino C, Krishnan S, Karanasos K, et al. Hydra: A federated resource manager for data-center scale analytics[C]//Proc of 16th USENIX Symp on Networked Systems Design and Implementation. Berkeley, CA: USENIX Association, 2019: 177−192

[8] Wang Zhijun, Li Huiyang, Li Zhongwei, et al. Pigeon: An effective distributed, hierarchical datacenter job scheduler[C]//Proc of the ACM Symp on Cloud Computing. New York: ACM, 2019: 246−258

[9] Rosenkrantz S, Li Huiyang, Enganti P, et al. JADE: Tail-latency-SLO-aware job scheduling for sensing-as-a-service[C]//Proc of 2020 IEEE/ACM the 13th Int Conf on Utility and Cloud Computing. Piscataway, NJ: IEEE, 2020: 366−373

[10] Dahlin M. Interpreting stale load information[J]. IEEE Transactions on Parallel and Distributed Systems, 2002, 11(10): 1033−1047

[11] Dean J, Barroso L A. The tail at scale[J]. Communications of the ACM, 2013, 56(2): 74−80

[12] Hellerstein J L, Diao Yixin, Parekh S, et al. Feedback Control of Computing Systems[M]. New York: John Wiley & Sons, 2004

[13] Tassiulas L, Ephremides A. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks[C]//Proc of 29th IEEE Conf on Decision and Control. Piscataway, NJ: IEEE, 1990: 2130−2132

[14] Neely M. Stochastic Network Optimization with Application to Communication and Queueing Systems[M]. San Rafael: Morgan & Claypool Publishers, 2010

[15] Reiss C, Tumanov A, Ganger G R, et al. Heterogeneity and dynamicity of clouds at scale: Google trace analysis[C]//Proc of the 3rd ACM Symp on Cloud Computing. New York: ACM, 2012: 1−13

[16] Wang Weitao, Wu Xinyu Crystal, Tammana P, et al. Closed-loop network performance monitoring and diagnosis with Spider-Mon[C]//Proc of the 19th USENIX Symp on Networked Systems Design and Implementation (NSDI 22). Berkeley, CA: USENIX Association, 2022: 267−285

[17] Tahir A, Alt B, Rizk A, et al. Load balancing in compute clusters with delayed feedback[J]. IEEE Transactions on Computers, 2022, 72(6): 1610−1622

[18] Shobhana B V, Narayana S, Nath B. Load balancers need in-band feedback control[C]//Proc of the 21st ACM Workshop on Hot Topics in Networks. New York: ACM, 2022: 76−84

[19] Mitzenmacher M. How useful is old information (extended abstract)?[C]//Proc of the 16th Annual ACM Symp on Principles of Distributed Computing. New York: ACM, 1997: 83−91

[20] Vvedenskaya N D, Dobrushin R L, Karpelevich F I. Queueing system with selection of the shortest of two queues: An asymptotic approach[J]. Problemy Peredachi Informatsii, 1996, 32(1): 20−34

[21] Azar Y, Broder A Z, Karlin A R, et al. Balanced allocations[C]//Proc of the 26th Annual ACM Symp on Theory of Computing. New York: ACM, 1994: 593−602

[22] Mitzenmacher M. The power of two choices in randomized load balancing[J]. IEEE Transactions on Parallel and Distributed Systems, 2002, 12(10): 1094−1104

[23] McFadden D. Conditional logit analysis of qualitative choice behavior[M]//Zarembka P. Frontiers in Econometrics. New York: Academic Press, 1972: 105−142

[24] Train K E. Discrete Choice Methods with Simulation[M]. Cambridge, UK: Cambridge University Press, 2009

[25] Hollot C V, Misra V, Towsley D, et al. Analysis and design of controllers for AQM routers supporting TCP flows[J]. IEEE Transactions on Automatic Control, 2002, 47(6): 945−959

[26] Chiu D M, Jain R. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks[J]. Computer Networks and ISDN Systems, 1989, 17(1): 1−14

[27] Kelly F P, Maulloo A K, Tan D K H. Rate control for communication networks: shadow prices, proportional fairness and stability[J]. Journal of the Operational Research Society, 1998, 49(3): 237−252

[28] Beyer B, Jones C, Petoff J, et al. Site Reliability Engineering: How Google Runs Production Systems[M]. Sebastopol: O'Reilly Media, 2016

[29] Cho I, Saeed A, Fried J, et al. Overload control for µ-scale RPCs with breakwater[C]//Proc of the 14th USENIX Symp on Operating Systems Design and Implementation (OSDI' 20). Berkeley, CA: USENIX Association, 2020: 299−314

[30] Cho I, Saeed A, Park S J, et al. Protego: Overload control for applications with unpredictable lock contention[C]//Proc of 20th USENIX Symp on Networked Systems Design and Implementation (NSDI' 23). Berkeley, CA: USENIX Association, 2023: 725−738

[31] Park J, Park J, Jung Y, et al. TopFull: An adaptive top-down overload control for SLO-oriented microservices[C]//Proc of the ACM SIGCOMM 2024 Conf. New York: ACM, 2024: 876−890

[32] Xing Jiali, Giannoukos A, Loh P, et al. Rajomon: Decentralized and coordinated overload control for latency-sensitive microservices[C]//Proc of the 22nd USENIX Symp on Networked Systems Design and Implementation (NSDI 25). Berkeley, CA: USENIX Association, 2025: 21−36

[33] Bhattacharya R, Gao Yuan, Wood T. Dynamically balancing load with overload control for microservices[J]. ACM Transactions on Autonomous and Adaptive Systems, 2024, 19(4): 1−23

[34] Agrawal K, Abdu Jyothi S. Cooperative graceful degradation in containerized clouds[C]//Proc of the 30th ACM Int Conf on Architectural Support for Programming Languages and Operating Systems (Volume 1). New York: ACM, 2025: 214−232

[35] Tang Xuhao, Liu Fagui, Wang Bin, Li Chao, Jiang Jun, Tang Quan, Chen Weiming, He Fengwen. Survey on Task Scheduling in Inter-Cloud Environment[J]. Journal of Computer Research and Development, 2023, 60(6): 1262−1275
(唐续豪, 刘发贵, 王彬, 等. 跨云环境下任务调度综述[J]. 计算机研究与发展, 2023, 60(6): 1262−1275)

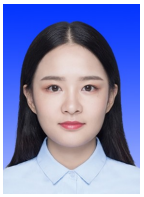
[36] Verma A, Pedrosa L, Korupolu M, et al. Large-scale cluster management at Google with Borg[C]//Proc of the 10th European Conf on Computer Systems. New York: ACM, 2015: 1−17

[37] Lo D, Cheng L, Govindaraju R, et al. Heracles: Improving resource efficiency at scale[C]//Proc of the 42nd Annual Int Symp on Computer Architecture. New York: ACM, 2015: 450−462

[38] Delimitrou C, Kozyrakis C. Paragon: QoS-aware scheduling for heterogeneous datacenters[J]. ACM SIGPLAN Notices, 2013, 48(4): 77−88

[39] Delimitrou C, Kozyrakis C. Quasar: Resource-efficient and QoS-aware cluster management[J]. ACM SIGPLAN Notices, 2014, 49(4): 127−144

[40] Jacobson V. Congestion avoidance and control[J]. ACM SIGCOMM Computer Communication Review, 1988, 18(4): 314−329



Zhou Yuwen, born in 1998. PhD candidate. Her main research interest includes computility networks.

周钰雯, 1998 年生。博士研究生。主要研究方向为算力网络。



Ren Bangbang, born in 1994. Associate professor. His main research interests include software-defined networking and network optimization.

任棒棒, 1994 年生。副教授。主要研究方向为软件定义网络、网络优化。



He Rui, born in 1998. PhD candidate. His main research interests include computility networks and network optimization.

何　瑞, 1998 年生。博士研究生。主要研究方向为算力网络、网络优化。



Guo Deke, born in 1980. Professor. His main research interests include distributed systems, software-defined networking, data center networking, wireless and mobile systems, and interconnection networks.

郭得科, 1980 年生。教授。主要研究方向为分布式系统、软件定义网络、数据中心网络、无线移动系统、互连网络。