面向算力网络的端网协同RDMA拥塞控制

刘亚萍[1,2]，严定宇[2,3]，方滨兴[1]，许名广[2]，张硕[1,2]，杨智凯[1]
（1.广州大学网络空间安全学院，广东 广州 510006；2.鹏城实验室，广东 深圳 518108；
3.北京邮电大学可信分布式计算与服务教育部重点实验室，北京 100876）

摘　要：为解决远程直接内存访问（RDMA）技术跨域互联场景下的长控制回路及混合流量拥塞问题，提出了一种面向算力网络的拥塞控制方法WRCC。采用基于输入速率的公平速率计算策略，由交换机精确计算拥塞队列的端口公平速率。结合近源交换机双控制回路与带内网络遥测技术，实现端网协同的速率控制，快速响应拥塞。仿真实验表明，与现有商用方法相比，WRCC能将平均流完成时间降低8%~47%，还能将尾流完成时间降低10%~70%。原型系统测试表明，与英伟达CX7相比，WRCC将短距离场景下尾时延降低7%~49%。在640 km长距离场景下，WRCC将平均时延降低2%~7%，尾时延降低45%~49%，平均吞吐量提升26%~90%。

关键词：拥塞控制；远程直接内存访问；算力网络；端网协同
中图分类号：TP393.0
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2026038

Congestion control for RDMA with end-network collaboration in computing power network

Liu Yaping[1,2], Yan Dingyu[2,3], Fang Binxing[1], Xu Mingguang[2], Zhang Shuo[1,2], Yang Zhikai[1]
1. Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China
2. Pengcheng Laboratory (PCL), Shenzhen 518108, China
3. Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Ministry of Education, Beijing 100876, China

Abstract: To address the long control loop and hybrid traffic congestion issues caused by cross-domain interconnection scenarios of remote direct memory access (RDMA) technology, a congestion control method for computing power networks, named WRCC (WAN RDMA congestion control), was proposed. A fair rate computing strategy based on input rate was employed, enabling switches to accurately calculate the port fair rate of congested queues. Combined with dual control loops on the near-source switch and in-band network telemetry technology, it achieved end-network collaboration rate control and rapidly responded to congestion. Simulation experiments demonstrate that compared with existing commercial methods, WRCC reduces the average and tail flow completion time by 8%~47% and 10%~70%. Prototype system tests indicate that compared with NVIDIA CX7, WRCC reduces the tail latency by 7%~49% in short-distance scenarios. In long-distance scenarios of 640 kilometers, WRCC reduces the average and tail latency by 2%~7% and 45%~49%, while achieving the average throughput improvement of 26%~90%.

Keywords: congestion control, remote direct memory access, computing power network, end-network collaboration

收稿日期：2025–12–11；修回日期：2026–02–02
通信作者：刘亚萍，ypliu@gzhu.edu.cn
基金项目：新一代人工智能国家科技重大专项基金资助项目（No.2025ZD0122203）
Foundation Item：The New Generation Artificial Intelligence-National Science and Technology Major Project (No.2025ZD0122203)

0 引言

随着人工智能（artificial intelligence, AI）与大模型训练等技术的快速发展，由多个智算中心构成的算力网络通过协同调度计算与网络资源，将成为未来支撑AI应用的关键基础设施，其重要性日益凸显[1]。国内外公司对智算中心AI任务的测试结果显示，AI任务的完成时间平均30%（依据不同AI模型，从18%~57%不等）用于网络传输，网络传输开销已成为除算力之外的第二大瓶颈点[2]。算力网络不仅包括智算中心内网络，还涵盖由多个基于长距离链路和智算中心互联（data center interconnect, DCI）交换机（switch, SW）组成的跨智算中心网络[3]。远程直接内存访问（remote direct memory access, RDMA）技术因其零拷贝和内核旁路能力被广泛部署在智算中心内网络中[4-5]。随着跨智算中心应用场景的出现，RDMA技术也正在被扩展应用到跨智算中心网络，以实现高性能的跨域数据传输。例如，Microsoft Azure在其跨智算中心网络中部署了RDMA技术，以增强跨域存储流量性能[6]。

跨智算中心网络通过长距离RDMA技术能扩展智算中心规模，为社会大规模广泛的AI算力服务奠定了技术基础。一方面，受限于电力，单个智算中心的规模限制了其智算中心内最大的计算与存储资源，Facebook、谷歌等云服务提供商倾向于通过跨智算中心网络实现计算和存储资源的共享[7]。另一方面，现有智算中心内应用通常使用RDMA技术，通过RDMA技术可以保证跨域应用接口的一致性，降低跨智算中心应用部署的难度[8-10]。此外，专用DCI交换机使运营商能够部署专用协议以满足RDMA技术对于网络的要求[6,11]。

目前智算中心内AI集群中普遍使用的RDMA网络是基于融合以太网的RDMA（RDMA over converged Ethernet version 2, RoCEv2）网络，它与现有的以太网基础设施无缝集成，并保留了RDMA技术的高吞吐量和低时延优势[5,12]。RoCEv2网络的高性能传输依赖于基于优先级流量控制（priority-based flow control, PFC）方法构建的无损以太网，PFC方法支持网络中拥塞节点向上游设备端口发送暂停帧以缓解网络拥塞，但这可能会引发一系列网络问题，如队头阻塞、PFC帧风暴等[4-5]，现有解决方案是通过流级别的拥塞控制来缓解这些性能瓶颈。

然而，研究表明RoCEv2网络的拥塞控制在跨智算中心网络中的部署面临着许多挑战[9,13]，主要挑战如下。1) 长控制回路问题。跨智算中心网络拥有更大的基础往返时延（round-trip time, RTT），长距离流量的长控制回路使现有端到端拥塞控制方法难以及时管理跨域线速突发流量，导致智算中心内交换机缓冲区过度占用。2) 混合流量拥塞问题。当长距离突发流量与智算中心内突发流量发生碰撞时，会出现混合流量拥塞。长距离流量对混合拥塞的反应较为迟缓，而智算中心内流量反应迅速，传统端到端拥塞控制方法（如DCQCN[10]和HPCC[12]）难以同时准确控制跨域流量和智算中心内流量的速率收敛，这加剧了网络拥塞并导致长交换机队列。

本文提出了一种广域网RDMA拥塞控制（WAN RDMA congestion control, WRCC）方法。为解决长控制回路在跨域流量突发时导致的速率控制较慢问题，WRCC采用基于双控制回路的流量速率控制方法，快速响应跨智算中心网络下的流量突发。WRCC提出了基于队列输入速率的公平速率计算策略，动态地为每个拥塞端口计算流量公平速率，以缓解混合流量拥塞导致的速率收敛问题。WRCC借助带内网络遥测（inband network telemetry, INT）技术[14]，通过嵌入每个数据包的固定长度INT头部实现流级别速率控制。其公平速率计算仅依赖于基本的流属性（速率和时延），具有广泛的适用性，并消除了配置复杂附加参数（如参考队列长度或目标带宽利用率等）的需要。

综上所述，本文的主要创新点如下。

1) 针对长控制回路问题，提出了一种基于双控制回路的流量速率控制方法，快速响应跨智算中心网络下的流量突发。其中，亚RTT交换机控制回路精准识别并主动降低突发流量和跨域流量的发送速率，端到端控制回路则对常规流量进行速率调节。

2) 针对混合流量拥塞问题，提出了一种基于输入速率的端口公平速率（port fair rate, PFR）计算方法，由网内交换机主动为不同RTT流量分配公平速率，并采用基于拥塞队列输入速率的精准速率调整策略，感知端口实际拥塞程度和剩余带宽，准确调整公平速率。

3) 基于400 Gbit/s现场可编程门阵列（field-programmable gate array, FPGA）网卡和可编程交换机实现了WRCC原型系统。在640 km长距离环境

下进行了原型系统的测试，并进行了算法的大规模仿真。仿真实验表明，WRCC通过维持较低的交换队列长度，将整体网络端口暂停时间减少一个数量级，实现了更短的整体流完成时间（flow completion time, FCT）。与现有DCQCN、HPCC等典型算法相比，WRCC显著降低了整体FCT，将平均FCT降低8%~47%，尾FCT降低10%~70%。640 km长距离环境下的原型系统实验结果表明，与支持DCQCN算法的英伟达CX7相比，WRCC将平均时延降低2%~7%，尾时延降低45%~49%，平均吞吐量提升26%~90%，有效缓解了跨智算中心网络流量突发问题。

1　相关工作

1) 智算中心内拥塞控制。DCQCN[10]利用基于交换机队列长度的显式拥塞通知（explicit congestion notification, ECN）进行拥塞标记，端侧据此调整发送速率。Timely[15]和Swift[16]则是两种基于RTT的解决方案，其优势在于不需要交换机支持。HPCC[12]采用了更精细的INT机制，直接利用交换机端口带宽利用率信息来精准控制速率。DCQCN与HPCC已获得业界认可并被广泛部署。ACC[17]通过确认ACK报文与精准的流标记方法来调整速率。而RoCC[18]允许交换机根据固定队列长度计算公平速率，并主动通知发送方降速。然而，上述方案主要针对单个智算中心网络设计，难以有效解决跨域场景下因长距离传输带来的长控制回路和混合流量拥塞问题。

2) 跨智算中心拥塞控制。现有跨智算中心TCP拥塞控制策略通常基于流量时延特征区分短距与长距流量，并实施差异化控制以优化混合流量性能。GEMINI[13]结合ECN与RTT作为复合拥塞信号，并为不同RTT的流量设置参数。GTCP[19]在主机端根据时延差异采用不同策略。Annulus[20]专注于缓解近源交换机拥塞，它借助机架顶部交换机快速检测混合拥塞，并通过双控制回路抑制长距离流量，但远端拥塞仍需依赖端侧策略。此外，由于RDMA拥塞控制与底层硬件深度耦合，这些基于时延差异的TCP难以直接应用于RDMA网络。

对于跨智算中心RDMA拥塞控制，BiCC[9]和LSCC[21]通过DCI交换机的快速通知报文来缩短长控制回路。然而，DCI交换机难以精准感知远端智算中心内部的混合拥塞状态，且其仍依赖于端到端协议，因而无法完全解决混合流量场景下的速率收敛问题。与BiCC仅将策略部署于DCI交换机不同，本文方法在跨智算中心的全路径上，由交换机实时监测混合流量状态，并主动为拥塞流分配公平速率，不需要依赖端侧启发式速率调整机制。这不仅能够快速响应近端拥塞，避免PFC暂停，还能精准识别远端智算中心的拥塞状况，进而实现对不同RTT混合流量的速率控制。

3) 跨智算中心流量控制。Swing[8]通过将流量控制信号与数据包缓冲区解耦，解决了跨域PFC时延问题。Bifrost[22]则使用传输中的数据包和排队中的数据包来实现精准流量暂停。这些方法旨在减少DCI交换机的队列长度，同时保证无损网络，与本文提出的跨智算中心网络拥塞控制方法可相互补充。

2　研究动机

本节首先阐述跨智算中心网络中长距离RDMA传输所引发的潜在问题。通过典型实验揭示现有商用拥塞控制方法在该场景下的实际性能局限。针对上述问题，本节提出了WRCC的设计思路。

2.1　跨智算中心网络及其挑战

跨智算中心网络及其应用。图1展示了一种典型的智算中心互联系统，每个智算中心通过DCI交换机与其他智算中心相连[7,23-24]。云服务商通过长距离链路、DCI交换机/路由器或区域级网关将不同智算中心进行互联，以满足不断增长的云服务规模需求[7]。此外，云服务商采用跨智算中心数据备份应用以抵御灾难级园区级故障，其备份流量往往是周期性长流量[23,25]。随着大语言模型的发展，分布式模型训练也需要利用不同智算中心内的计算和存储资源[26]。

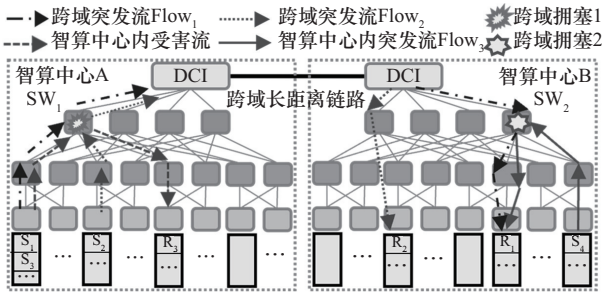


图1　跨智算中心网络和跨域拥塞挑战

多对一（Incast）流量模式及其问题。智算中心AI应用程序通常会产生多对一流量模式，例如，

参数服务器架构等分布式机器学习通信架构需要所有节点同时进行梯度同步操作，这类Incast流量模式易引发长尾时延问题[27-28]。在跨智算中心RoCEv2网络中，Incast流量模式与跨域流量结合会造成许多新问题。

1) 长控制回路问题。如图1所示，跨域突发流$Flow_1$（S_1→R_1）和$Flow_2$（S_2→R_2）在智算中心A的SW_1处引发Incast拥塞（跨域拥塞1）。端到端拥塞控制策略需要一个跨域RTT（数毫秒以上）来检测该拥塞。在此期间，SW_1会经历持续拥塞。为了保持网络无损，$Flow_1$的入端口会经历周期性暂停，导致沿相同路径的智算中心内受害流（S_3→R_3）性能下降。

2) 混合流量拥塞问题。跨域突发流$Flow_1$（S_1→R_1）和智算中心内突发流$Flow_3$（S_4→R_1）导致智算中心B中SW_2出现跨域Incast拥塞（跨域拥塞2）。跨域突发流$Flow_1$对该拥塞的反应较慢，需要跨域RTT才能使其速率调整在远程SW_2处生效。相比之下，智算中心内突发流$Flow_3$对拥塞的响应更快。这种响应时间的差异使发送方S_1难以准确评估跨域拥塞状态，从而导致错误的收敛点。而跨域流量错误收敛则会导致队列变长，降低整体网络性能。

2.2　现有拥塞控制方法面临的挑战

RoCEv2网络的高性能传输依赖于基于PFC的无损网络。PFC支持网络中拥塞节点暂停上游设备数据传输以缓解网络拥塞，但可能会引发队头阻塞、PFC帧风暴等问题。现有解决方案是通过流级别的拥塞控制来缓解这些性能瓶颈。然而，在跨智算中心网络下，RDMA传输会面临长控制回路与混合流量拥塞的新问题。

本文评估了HPCC和广泛部署的DCQCN算法在跨域Incast场景（跨域拥塞2）下的性能表现。仿真实验的拓扑如图1所示，其中跨域长距离链路往返时延为2 ms，其余配置与第4节一致。实验结果揭示了现有方法难以应对新的挑战。

1) 长控制回路难以准确控制跨域流量。传统的拥塞控制方法采用端到端拥塞信号（如DCQCN的ECN信号和HPCC的INT信号），而跨域长距离链路显著增加了跨域流量的反馈时延。如图2(a)所示，DCQCN的拥塞通知报文需要经过一个跨域RTT才能被发送方S_1接收，其降速效果需要跨域RTT后才能在SW_2处生效。这导致SW_2发生拥塞时，跨域突发流$Flow_1$延迟一个RTT才能感知拥塞并降速，而智算中心内突发流$Flow_3$则能够迅速响应拥塞，导致两种流量出现异步竞争问题。此外，发送方S_1长控制回路不能及时感知远程SW_2的拥塞状态，导致其跨域突发流$Flow_1$未能正确收敛，智算中心内突发流$Flow_3$也受$Flow_1$影响，存在收敛问题。

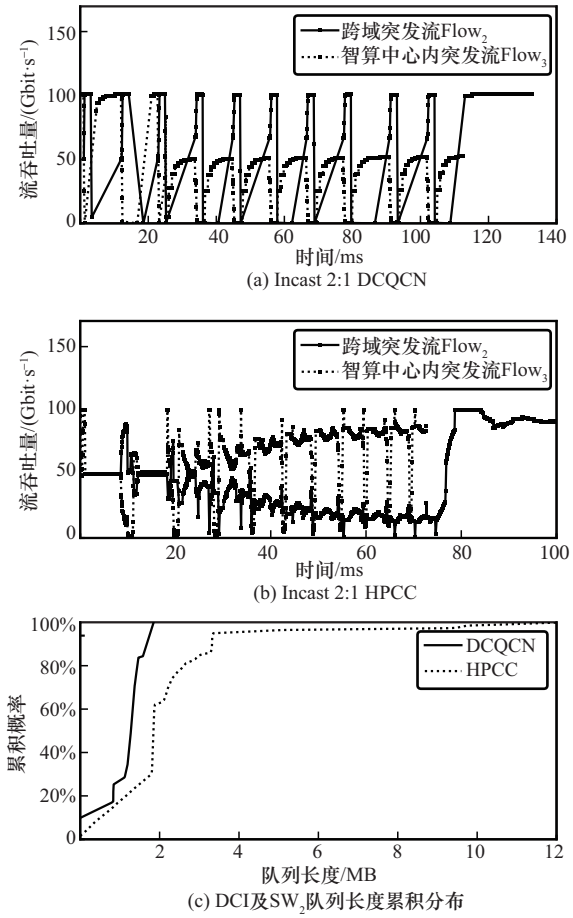


图2　跨域拥塞2下的流吞吐量和交换机队列分布

如图2(b)所示，HPCC的接收方R_1将INT拥塞信号捎带在ACK中通知发送方降速，ACK需要经历一个跨域RTT才能到达发送方S_1。这导致HPCC在跨域场景下同样面临跨域流量速率收敛和公平性挑战。

2) 混合流量拥塞导致缓冲区过度占用。如图2所示，DCQCN和HPCC在混合拥塞下，跨域突发流$Flow_1$和智算中心内突发流$Flow_3$长时间无法正确收敛，这导致交换机形成更长的队列。如图2(c)所示，DCQCN的70%队列长度超过1 MB，而HPCC

的 70% 队列长度超过 1.8 MB，队列长度最大为 12 MB。这种混合拥塞下的长队列会导致其他流量的可用缓冲区减少并增加整体网络时延。

2.3　设计目标与思路

基于上述观察，本文提出的跨智算中心网络RDMA拥塞控制应具备以下设计目标。

1) 精准识别跨域流量。跨域长流量会导致智算中心内交换机缓冲区高度占用，拥塞控制应迅速识别端口流量的混杂情况，及时降低突发跨域流量的速率，以避免网络拥塞加剧。

2) 快速响应拥塞。新的拥塞控制方法应尽量缩短拥塞控制回路，以快速响应跨域长距离链路下的复杂拥塞。

WRCC设计思路为：1) 与大规模部署的HPCC相似，WRCC利用INT技术并对其进行扩展，使数据包携带流量的基本信息（如智算中心标识、发送速率等），由交换机准确识别影响整体网络性能的突发流量和跨域流量；2) WRCC支持近源交换机直接生成拥塞通知报文，并降低跨域流量速率。

3　WRCC设计

本节首先对WRCC拥塞控制机制的整体框架进行概述。然后，详细阐述WRCC关键算法机制与原理。最后，对WRCC的稳定性进行分析，并提出关键参数的自动调整机制，以确保其在复杂网络环境下的高效运行与性能优化。

3.1　设计概述

WRCC架构如图3所示。为解决混合流量拥塞问题，WRCC采用基于INT和PFR的速率调整机制。WRCC在每个数据包中扩展了一个INT头部，包含传输速率、RTT估计值和域标识符等基本信息。交换机在转发数据包时，收集端口流量统计信息，并根据流量状态计算PFR，主动为不同类型流量分配公平速率，从而缓解混合拥塞导致的不公平和难收敛问题。为解决长控制回路问题，WRCC采用双控制回路系统，包括端到端控制回路和亚RTT交换机控制回路。

1) 端到端控制回路。发送方在发送数据时，将基本信息（如传输速率等）附加到INT头部中。交换机在收到报文后，将流速率与PFR进行比较分析。若流速率超过PFR，则将原始速率替换为PFR。随后，接收方在ACK中捎带网络内公平速率供发送方调速。

2) 亚RTT交换机控制回路。WRCC利用近源交换机（与发送方位于同一智算中心）快速解决近源拥塞。当检测到突发流速率超过PFR时，近源交换机立即将带有PFR的拥塞通知消息发回发送方，以快速降低跨域突发流量的速率，提高网络整体响应速度。

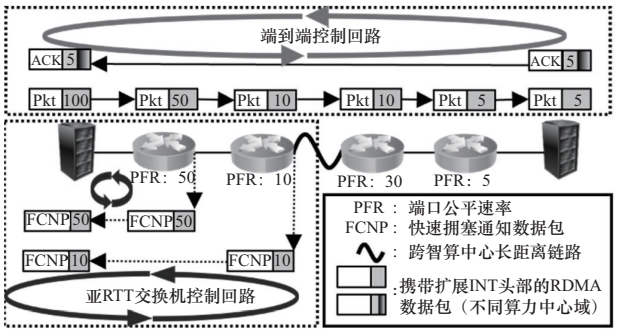


图3　WRCC架构

针对长控制回路问题，所提出的基于双控制回路的流量速率控制方法在3.1节中详细描述。WRCC通过在可编程交换机数据平面实现INT头部解析与快速拥塞响应机制，构建双控制回路系统。针对混合流量拥塞问题，所提出的基于输入速率的PFR计算方法在3.3节至3.5节中阐述。WRCC通过在RoCEv2数据包中加入扩展INT头部以支持基于INT和PFR的精准速率调整，并详细说明PFR的计算步骤，以及网卡端侧的INT信息处理及其速率调整步骤。最后对WRCC方法进行稳定性的理论分析，并推导出关键参数自动调整策略。

3.2　基于双控制回路的流量速率控制方法

基于双控制回路的流量速率控制方法工作原理如图4所示。WRCC通过在交换机数据平面执行信息统计、INT头部解析与修改、快速拥塞通知数据包（fast congestion notification packet, FCNP）判断与生成，实现基于双控制回路的流量速率控制。1) 数据平面统计（①和②），基于交换机Ingress阶段的转发元信息，维护端口间转发统计表，用于后续的PFR计算；2) INT头部解析与修改（③），该功能主要在Egress阶段，解析每个数据包INT头部中的公平速率信息，并根据PFR对INT头部执行修改；3) FCNP判断与生成（④和⑤），解析数据包INT头部的域标识符和FBDP字段，并为突发流量生成FCNP以快速响应网络拥塞。

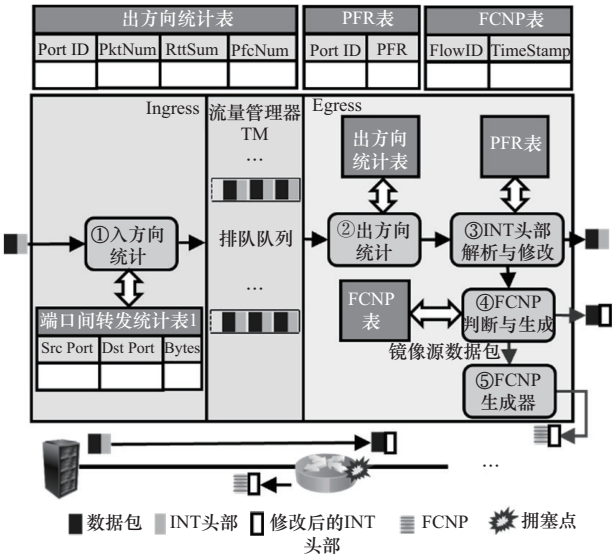


图4　基于双控制回路的流量速率控制方法工作原理

数据平面统计。WRCC采用基于端口输入速率的公平速率计算方法（详见第3.4节）。数据平面需统计端口间数据包转发信息，并实时更新端口间转发统计表。当数据包被转发至流量管理器时，利用数据包元信息中的源/目的端口号作为索引，将数据包长度信息添加到端口间转发统计表的转发字节数目Bytes字段。

为了精准识别出端口的混合流量情况，WRCC在Egress阶段解析数据包INT头部的RTT字段，并将其添加到出方向统计表的RttSum字段中，同时记录统计周期内数据包数目PktSum。交换机据此计算统计周期内出端口的平均RTT，作为识别各端口流量混合的主要指标。此外，无损网络中PFC暂停会降低端口队列输入速率，这表明严重网络拥塞的发生。因此，WRCC还在Egress阶段统计入端口的PFC暂停帧发送数目PfcNum，记录入端口的PFC暂停情况，用于感知PFC暂停的快速降速。

INT头部解析与修改。当数据包转发时，交换机根据出端口号检索PFR表，判断转发数据包流速率与出端口的PFR的大小关系。当流速率小于PFR时，则不修改数据包INT头部，直接转发；当流速率大于PFR时，则将PFR写入数据包的INT头部公平速率字段，并进行后续FCNP生成与判断。

FCNP判断与生成。交换机读取INT头部的域标识符和FBDP字段进行判断。若交换机与数据包的域ID相同且FBDP大于0（即突发拥塞发生在近源交换机处），则交换机执行镜像操作，并生成FCNP；否则，证明该拥塞为远程拥塞，仅将修改后的INT头部数据包进行简单转发。

FCNP生成步骤如下。首先对源数据包进行镜像操作，截断镜像包的RDMA头部，然后对镜像包进行修改将其转化为FCNP，具体包括IP地址和端口的源/目的翻转、设置FBDP字段标识FCNP、设置INT的数据包序列号（packet sequence number, PSN）等操作。

为防止同一流生成多个FCNP，交换机在生成FCNP后会对流的五元组进行哈希生成流ID，并将该流ID与当前FCNP时间戳保存在FCNP表中，仅当距离上次FCNP的时间大于固定时间间隔（WRCCT）时，才会生成下一个FCNP。

3.3　基于INT的RoCEv2数据包扩展

如图5(a)所示，WRCC通过附加INT字段来增强RoCEv2数据包和ACK的头部信息，主要包含以下4个字段。

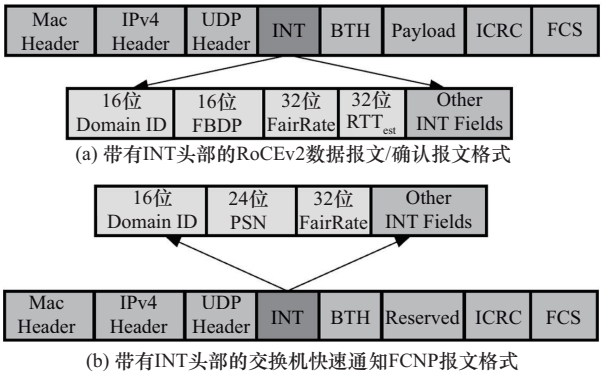


图5　带有扩展INT头部的RoCEv2报文格式

1) Domain ID（16位）：发送方和接收方所在智算中心实体的唯一标识符。

2) FBDP（16位）：突发流量标识符，表示发送方刚开始发送流量但未收到ACK。

3) FairRate（32位）：存储以1 Mbit/s为单位的标准化流速率，初始值为该数据包对应流的发送速率，中间交换机将路径中PFR写入该字段，用于传递网内最小公平速率。

4) RTT_{est}（32位）：表示RTT的估计值（以微秒为单位），作为交换机公平速率计算的参数，用于量化不同RTT流量的混合情况。

如图5(b)所示，WRCC支持近源交换机直接生成FCNP，以告知发送方迅速降速，主要包含以下3个字段。

1) Domain ID（16 位）：生成快速拥塞通知报文的交换机实体所在智算中心的唯一标识符。

2) PSN（24 位）：标识触发快速拥塞通知包的原RoCEv2数据包序号，避免端侧对同一数据包的降速信号重复响应。

3) FairRate（32 位）：标识交换机拥塞端口的PFR，单位为1 Mbit/s，用于告知发送方快速降速。

3.4　基于队列输入速率的端口公平速率计算方法

WRCC采用基于交换机的公平速率计算方法，由交换机周期性地计算拥塞端口公平速率并更新交换机PFR表。该方法将拥塞队列的队列输入速率作为计算参数输入，主要考虑以下因素。

1) 队列输入速率反映端口拥塞水平。相较于固定的队列长度，队列输入速率更能反映端口的拥塞水平，并能计算合适的调速幅度[29]。如图6所示，当Incast 8:1拥塞发生时，出端口的输入速率会维持在800 Gbit/s，通过感知该队列的输入速率，可精准计算降速幅度（SlowDown为$\frac{1}{8}$）。当带宽利用率不足时，也能在不拥塞的情况下精准调整增速幅度。

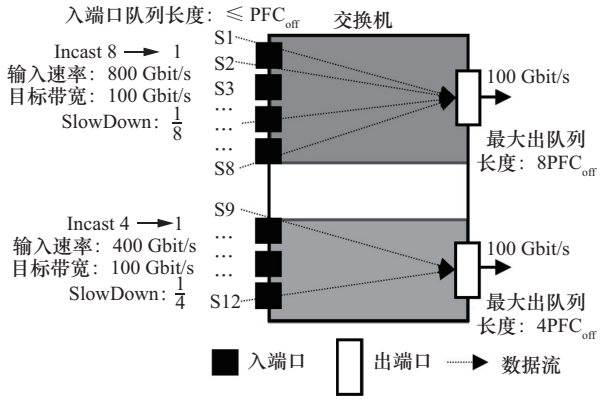


图6　基于队列输入速率的乘性降速

2) PFC限制下的队列长度管理。由于PFC的限制，不触发PFC暂停的最大入端口队列长度为PFC_{off}。通过统计输入流量的入端口数目，可计算出端口的最大队列长度MaxQueue，当出端口队列长度接近该最大队列长度时，可以在不触发PFC的情况下充分利用交换机缓冲区，并支持短暂的流量突发情况。

交换机每隔一个周期（WRCCT）会根据各个统计信息表更新每个端口的PFR表。考虑到交换机控制与数据平面存在较大的时延，公平速率的计算与更新主要依赖于交换机数据平面执行，可以通过基于ASIC的硬件或基于移位的近似算法实现公平速率计算。该计算方法主要包括两个步骤：表数据处理和PFR计算。WRCC在交换机算法中使用的关键符号如表1所示。

表1　　交换机算法关键符号

参数	含义
F_{min},F_{max}	最小/最大端口公平速率
WRCCT	交换机公平速率采样周期
$RateRatio_{max}$	精准调速阶段最大比率
$QRatio_{max}$	最大队列长度比率
B_{Tar}	交换机端口目标带宽
R_{Fair}	交换机端口公平速率
Q_p,Q_{max}	端口队列长度及最大队列长度
Q_{Extra}	端口队列超限长度
R_{input}	端口队列输入速率
RTT_{avg}	出端口平均RTT
SlowDown	乘性速率降低因子
Sum_{PFC}	入端口PFC触发计数
α,β	精准调速因子

3.4.1　表数据处理

对于任意出端口p，按照式(1)计算出端口p的公平速率计算参数。

$$
\begin{aligned}
&I = i \in \text{SwitchPortID} \mid \text{Bytes}[i][p] > 0 \\
&\text{Sum}_{\text{PFC}} \leftarrow \sum_{i \in I} \text{PfcNum}[i] \\
&Q_{\max} \leftarrow \sum_{i \in I} \text{PFCth}[i] \quad \text{ALLBytes} \leftarrow \sum_{i \in I} \text{Bytes}[i][p] \\
&R_{\text{input}} \leftarrow \frac{\text{ALLBytes}}{\text{WRCCT}} \quad \text{AvgRTT} \leftarrow (1-\lambda)\,\text{AvgRTT}_{\text{last}} + \\
&\qquad \lambda \frac{\text{RttSum}[p]}{\text{PktNum}[p]}
\end{aligned} \tag{1}
$$

其中，集合I表示所有在WRCCT内向出端口p发送过数据的入端口号，根据端口间转发统计表中Bytes表项计算得出。根据集合I，计算端口p的输入速率（R_{input}）、关联入端口的PFC触发计数（Sum_{PFC}）和端口p的最大队列长度（Q_{max}）。PfcNum、RttSum和PktNum存放在出方向统计表中，而PFCth则表示每个入端口的PFC暂停阈值。对于统计周期内AvgRTT，通过指数加权移动平均公式计算，使该统计值更加平滑，推荐因子λ设置为较小的0.1。

3.4.2 端口公平速率计算

WRCC速率调整包括两个阶段：基于队列输入的乘性速率降低阶段和基于目标带宽的加性速率调整阶段，如算法1所示。

基于队列输入的乘性速率降低阶段。WRCC采用乘性降速策略，用于解决线速流量突发问题。如图6所示，当出现大规模Incast流量时，交换机内部会出现端口级Incast的情况。由于聚合的输入速率超过出端口带宽，出口端口出现拥塞。此时根据队列输入速率R_{input}与端口目标带宽B_{Tar}的比率精准地调整PFR。在无损网络下，PFC的触发会强制降低入端口的输入速率，使其近似等于输出带宽。在这种情况下，当出端口检测到任何关联入端口的PFC暂停时（即算法1第3~5行），WRCC至少将该出端口的公平速率减少一半，以响应入端口的PFC暂停触发。

基于目标带宽的加性速率调整阶段。当队列输入速率不超过目标速率的固定比率（$\text{RateRatio}_{\max}$）时，WRCC回退到加性速率调整阶段。式(2)概述了采用的核心算法，该算法的灵感来自TCP RCP[29]中使用的算法，主要功能是根据输入速率和目标速率之差来调整公平速率。

$$R_{\text{Fair}} \leftarrow R_{\text{Fair}}\left[1+\frac{\text{WRCCT}}{\text{RTT}_{\text{avg}}}\frac{\alpha(B_{\text{Tar}}-R_{\text{input}})-\beta\dfrac{Q_{\text{Extra}}}{\text{RTT}_{\text{avg}}}}{B_{\text{Tar}}}\right] \tag{2}$$

在式(2)中，WRCC通过引入一个与平均RTT直接相关的因子$\dfrac{\text{WRCCT}}{\text{RTT}_{\text{avg}}}$来增强混合RTT流量场景的鲁棒性。当跨域流量与智算中心内部流量混合时，出端口的平均流量RTT会远大于采样周期WRCCT，WRCC会根据该因子谨慎地提高公平速率，这可能会导致潜在的低端口利用率，但在跨智算中心无损网络下这种谨慎是必要的，可以有效减少端口PFC暂停的发生。考虑到零队列可能会引发吞吐量损失问题，WRCC还允许出口队列维护固定比率（$\text{QRatio}_{\max}$）的最大出端口队列长度，以充分利用缓冲区并吸收突发流量。建议$\text{QRatio}_{\max}$的设置大于10%且小于60%，以充分利用交换机缓冲区并降低排队时延。

算法1 WRCC端口公平速率计算方法

定义 最小流速率$F_{\min}$，最大流速率$F_{\max}$，统计周期WRCCT，最大队列比率$\text{QRatio}_{\max}$，精准调速阶段最大比率$\text{RateRatio}_{\max}$，目标端口带宽B_{Tar}以及端口p的队列长度Q_p，初始化额外队列长度$Q_{\text{Extra}} \leftarrow 0$以及原端口公平速率$R_{\text{Fair}} \leftarrow \text{PortFairRate}[p]$

输入 $p, R_{\text{input}}, \text{Sum}_{\text{PFC}}, Q_{\max}, \text{AvgRTT}, Q_p$

输出 更新$\text{PortFairRate}[p] \leftarrow R_{\text{Fair}}$

1) $\text{RTT}_{\text{avg}} \leftarrow \max(\text{AvgRTT}, \text{WRCCT})$

2) 乘性降速因子$\text{SlowDown} \leftarrow \dfrac{R_{\text{input}}}{B_{\text{Tar}}}$

3) if $\text{Sum}_{\text{PFC}} > 0$ then

4) $\quad \text{SlowDown} \leftarrow \max(\text{SlowDown}, 2)$

5) end if

6) if $Q_p \geqslant \text{QRatio}_{\max} Q_{\max}$ then

7) $\quad Q_{\text{Extra}} \leftarrow Q_{\max} - Q_p$

8) end if

9) if $\text{SlowDown} \geqslant \text{RateRatio}_{\max}$ then

10) $\quad R_{\text{Fair}} \leftarrow \dfrac{R_{\text{Fair}}}{\text{SlowDown}}$（乘性降速）

11) else

12) $\quad \alpha, \beta \leftarrow \text{AutoParm}(R_{\text{Fair}})$

13) $\quad$ 根据式(2)更新端口公平速率R_{Fair}

14) end if

15) $R_{\text{Fair}} \leftarrow \text{Clamp}(R_{\text{Fair}}, F_{\min}, F_{\max})$（限速）

16) 调参函数$\text{AutoParm}(R_{\text{Fair}}, \text{level} = 2)$

17) $\quad$ if $R_{\text{Fair}} < B_{\text{Tar}}$ then

18) $\quad\quad \text{Interval} \leftarrow B_{\text{Tar}} - R_{\text{input}}$

19) $\quad\quad$ while $\text{Interval} < \dfrac{B_{\text{Tar}}}{\text{level}}$ and $\text{level} < 64$ do

20) $\quad\quad\quad \text{level} \leftarrow 2\text{level}$

21) $\quad\quad$ end while

22) $\quad$ end if

23) $\quad \alpha \leftarrow \dfrac{1}{\text{level}}$ and $\beta = \dfrac{\alpha}{2}$

24) $\quad$ return α, β

3.5 端侧网卡机制

1) 发送端算法。如算法2所示，发送方以线速启动RoCEv2流量，实现高速传输。WRCC动态调整INT头部的FBDP字段，用以标识流量突发和跨域流量情况。在整个传输过程中，跨域流量均会触发近源交换机的快速拥塞通知机制（算法2第8行），从而降低跨域大流量对智算中心网络影响。对于连接RTT值的估算，可以在连接初始化过程中由驱动主动下发给硬件。

算法2　WRCC发送端

定义　端口带宽B_{send}，发送方域标识符SDI，连接的发送速率R_c，增速定时器STimer，超时周期T_1。初始化起始发送速率$R_c \leftarrow B_{\text{send}}$，突发标识符FBDP ← 1，最大处理包序号$\text{PSN}_{\text{last}} \leftarrow 0$，重置增速定时器STimer($T_1$)

1) 处理ACK过程HandleACK(ack)
2)　FBDP ← 0
3)　if ack.PSN > lastPSN then
4)　　$R_c \leftarrow$ ack.INT.R_{Fair}
5)　　$\text{PSN}_{\text{last}} \leftarrow$ ack.PSN
6)　end if
7)　if ack.INT.DomainID ≠ SDI then
8)　　FBDP ← 1
9)　end if
10) 处理FCNP过程HandleFCNP(fcnp)
11)　if $R_c >$ fcnp.INT.R_{Fair} then
12)　　$R_c \leftarrow$ fcnp.INT.R_{Fair}
13)　　$\text{PSN}_{\text{last}} \leftarrow$ fcnp.INT.PSN
14)　　重置增速定时器STimer(T_1)
15)　end if
16) 处理增速定时过程STimerExpire
17)　$R_c \leftarrow \max(2R_c, B_{\text{send}})$
18)　重置增速定时器STimer(T_1)
19) 填充INT头部过程FillPKTINT(pkt)
20)　$\text{RTT}_{\text{est}} \leftarrow$获得连接RTT估计值
21)　pkt.INT ← INT(SDI,FBDP,R_c,RTT_{est})

FCNP中包含触发RoCEv2数据包的PSN，这使发送方能够维护已响应最大PSN的记录，并仅响应具有更大PSN的ACK，从而避免发送方对同一拥塞点的重复响应问题（算法2第3行）。此外，发送方仅响应具有更小公平速率的FCNP（算法2第11行），而对于速率增加，则依赖于ACK中携带的网内最小公平速率，这有效避免了FCNP可能导致的速率振荡问题。为了充分利用交换机缓冲区，WRCC还支持发送方周期性地进行乘性增速（算法2第17行），并通过双控制回路缓解潜在的流量突发问题。

2) 接收端算法。接收端网卡接收带有INT头部的数据包后，将其携带的网内最小公平速率INT头部复制到ACK头部，并将接收方的域标识符写入ACK的INT头部，显式告知流量的跨域情况。随后，接收方返回ACK，通知发送方调整速率。

3.6　WRCC分析

1) 额外开销。WRCC的头部扩展会引入一定的额外开销。具体而言，每个RoCEv2数据包将增加约12 B的头部开销。在典型场景下，即最大传输单元（maximum transmission unit, MTU）为1 024 B时，该开销会导致吞吐量下降约1.2%。本文认为这一开销在可接受范围内。

2) 稳定性分析。为了确保流量的公平收敛性，WRCC主要取决于式(2)定义的精确速率调整阶段。对于具有相同往返时延d_0、总输入流$y(t)$和队列长度$q(t)$的N条流穿过带宽为C的单个瓶颈链路而言，整个系统可以通过式(3)进行系统建模。

$$\begin{aligned}\dot{R}(t) &= R(t)\left(\frac{\alpha(C-y(t))-\beta\dfrac{q(t)}{d(t)}}{Cd(t)}\right)\\ y(t) &= NR(t-d_0)\\ d(t) &= d_0+\frac{q(t)}{C}\end{aligned}\tag{3}$$

其中，$R(t)$表示端口的公平速率，$d(t)$表示t时刻的RTT。本文将动态队列长度进行线性化，即$\dot{q}(t)=y(t)-C$；当速率和队列长度都是时不变时，即$\dot{R}=0$和$\dot{q}=0$时，整个系统达到平衡，该系统在稳定点附近的线性化形式可以表示为

$$\begin{aligned}\delta\dot{q}(t) &= N\delta R\left(t-d_0\right)\\ \delta\dot{R}(t) &= -\frac{\alpha}{d_0}\delta R\left(t-d_0\right)-\frac{\beta}{Nd_0^2}\delta q(t)\end{aligned}\tag{4}$$

对于该系统的稳定参数求解，已有研究[29-30]证明，当参数α和β满足如图7所示的范围时，整个系统对于任意N、C和d_0均能保持稳定。

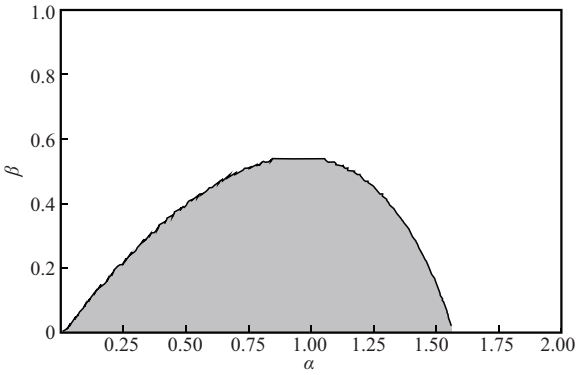


图7　Bode分析的参数稳定范围

3) 参数设置。WRCC的关键参数如表2所示。为了提高交换机统计精度，建议交换机采样周期WRCCT和FCNP生成周期设置为2~4倍智算中心最大RTT，以提高端口公平速率计算的准确性。相应地，发送方的增速周期T_1应大于或等于采样周期WRCCT，以便交换机能够及时控制发送方的乘性增速。式(2)中的加性速率调整阶段可以允许的最大输入速率是目标速率的两倍，$\mathrm{RateRatio}_{\max}$的取值范围为100%~200%。本文建议$\mathrm{RateRatio}_{\max}$设置为110%，$\mathrm{QRatio}_{max}$设置为50%，以增加网络对突发流量的鲁棒性，同时保持网络整体的高吞吐量。

表2 关键参数

实体	参数	含义
交换机侧	$F_{\min}$,$F_{\max}$	最小/最大端口公平速率
	$\mathrm{RateRatio}_{\max}$	精准调速阶段最大比率
	$\mathrm{QRatio}_{\max}$	最大队列长度比率
	WRCCT	交换机公平速率采样周期
	B_{Tar}	交换机端口目标带宽
网卡侧	STimer(T_1)	乘性增速定时器及其周期
	SDI, RDI	发送方/接收方域标识符
	B_{send}	发送方端口带宽

4) 自动参数调整。当出端口的队列输入速率不足时，应采用更大的α以快速增速；当输入速率接近目标带宽时，应采用较小的α以实现精准收敛。本文设计了一种自动参数调整机制，如算法1第15~23行所述，该机制根据输入速率和目标速率之差动态调节参数值，使目标参数满足上述稳定参数的范围，具体如式(5)所示。

$$\frac{1}{64} \leqslant \alpha \leqslant \frac{1}{2}, \beta = \frac{\alpha}{2} \tag{5}$$

4 实验评估

实验包括在NS3模拟平台[12]上的大规模模拟实验和通过Tofino2 400G可编程交换机DCS810上实现WRCC交换机算法和在400G Intel DK-DEV-AGI027的FPGA板卡上实现端侧算法，构建640 km的长距离原型系统实验。

4.1 大规模模拟实验

4.1.1 实验设置

1) 拓扑设置。本文基于NS3模拟平台[12]构建了一个基于FatTree架构[31]的典型跨智算中心网络拓扑，如图1所示。每个智算中心包含4个核心交换机、8个汇聚交换机和8个架顶交换机，每个架顶交换机连接4台服务器，每台服务器配备一张100 Gbit/s网卡。跨域长距离链路传播时延设置为1 ms[24]，智算中心内的传播时延为1 μs，智算中心内最大RTT为12 μs。智算中心内交换机之间的链路带宽为100 Gbit/s，不同DCI交换机之间链路带宽为400 Gbit/s。智算中心内交换机和DCI交换机缓冲区大小参考现有商业设备的缓冲区配置[9,12]分别为16 MB和250 MB。

2) 方法比较与参数设置。本文与现有商用拥塞控制方法DCQCN、HPCC以及最新的跨数据中心方法BiCC[9]进行性能对比。由于BiCC主要部署于DCI交换机，本文将其端侧核心算法配置为DCQCN。对于拥塞控制参数，DCQCN采用网卡供应商建议的默认配置[9-10]，HPCC则采用其论文推荐参数[12]。对于BiCC，DCI交换机虚拟输出队列遵循其论文参数配置[9]，设置为128。对于WRCC，默认场景下交换机采用周期（WRCCT）为24 μs，最大流速率（$F_{\max}$）为100 Gbit/s，最小流速率（$F_{\min}$）为100 Mbit/s。智算中心内交换机采用动态缓冲区并配置动态PFC阈值，对于DCI交换机的PFC参数配置，本文参考文献[8]配置静态大PFC阈值以保证跨域长距离链路吞吐量无损。

3) 工作负载。本文使用了3种开源智算中心流量分布，分别为网络搜索[32]、谷歌RPC[33]和数据挖掘[34]。在仿真实验中，跨智算中心网络的流量模式为All-to-all流量和Incast流量的混合。All-to-all流量根据上述开源流量分布随机生成，跨域流量与智算中心内流量的比例为5:1，符合实际应用场景[20]。Incast流量则是由跨域长距离链路带宽时延积大小的长流量组成，占跨域长距离链路的50%负载，模拟跨域网络场景下常见的数据备份流量。网络搜索采用默认智算中心流量模式。

4) 评价指标。本文主要采用以下性能评价指标：不同流量场景下的50%流完成时间（平均FCT）和99%流完成时间（尾FCT）及标准化流完成时间（FCT SlowDown）[12]，并将交换机端口暂停时间和交换机队列分布纳入性能评价指标，以全面衡量不同拥塞控制方法对跨智算中心网络中交换机缓冲区占用及端口暂停情况的影响。

4.1.2　模拟实验结果

在模拟实验中，首先评估默认参数（默认网络负载设置为60%）下的WRCC在不同流量模式下的性能表现。然后，调整网络负载大小和关键参数，评估WRCC的鲁棒性。

1) 基础结果

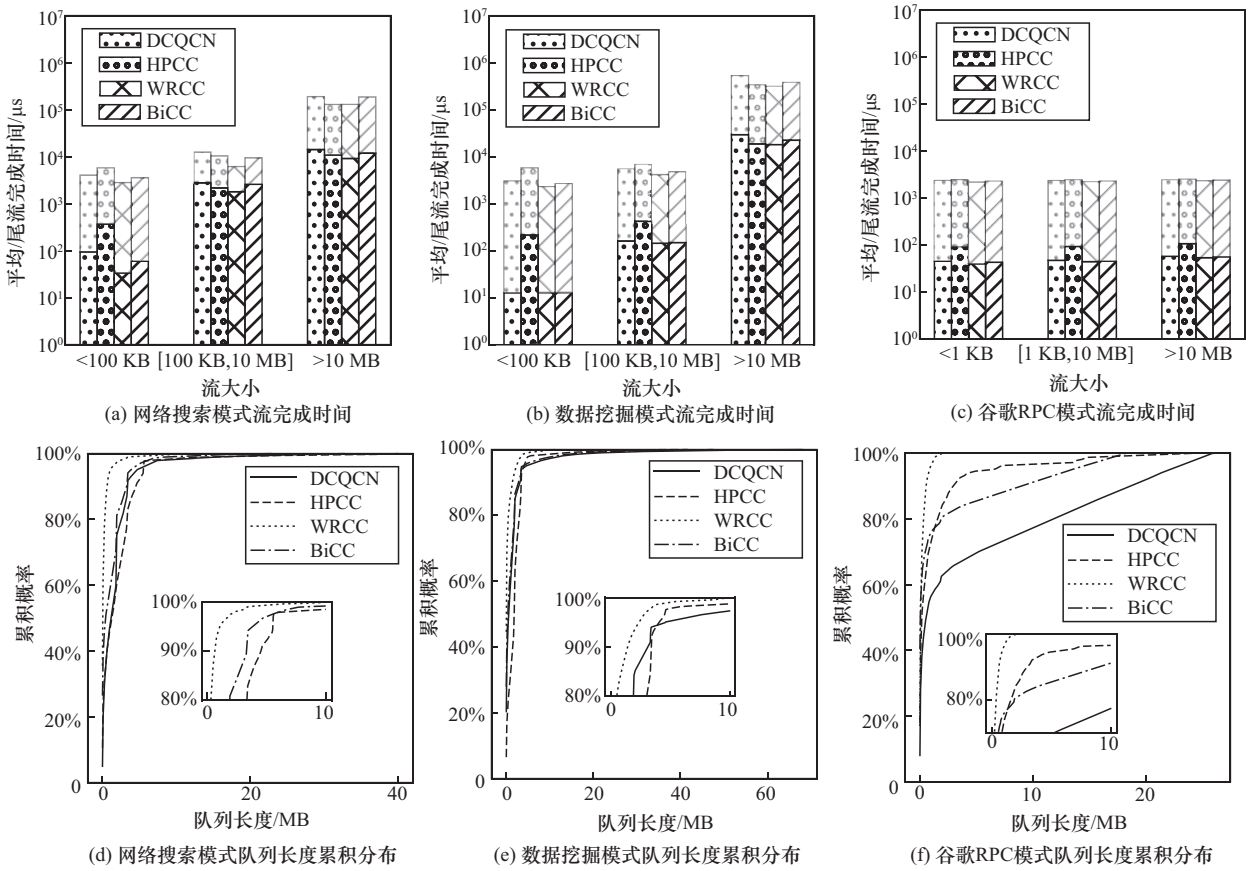
WRCC显著提升了整体流完成时间。图8是DCQCN、HPCC、BiCC和WRCC在不同流量模式下的平均/尾FCT及交换机队列长度累积分布。实验结果表明，WRCC在不同流量大小和流量模式下均表现出良好的性能提升。与DCQCN、HPCC和BiCC相比，WRCC最高能将短流的平均FCT分别降低63%、94%和43.8%，将短流的尾FCT分别降低30%、59%和21.5%。对于长流，WRCC最高能将平均FCT分别降低38.6%、49.7%和22.8%，将尾FCT分别降低41%、7.2%和30.3%。

WRCC对短流表现出更显著的性能提升。这种提升得益于双控制回路和交换机直接检测机制，能够在充分利用交换机缓冲区的同时维持较低队列长度，并在混合拥塞时减少队列长度，抑制跨域长流对交换机缓冲区过度占用问题，提高整体网络性能。

WRCC有效减少交换机队列长度。实验记录了交换机队列长度分布和交换机端口PFC暂停时间，如图8和图9所示。与DCQCN、HPCC和BiCC相比，WRCC可将第80百分位队列长度分别缩短40%~90%。WRCC在各种流量模式下能保持较低的队列长度，因此更不易触发PFC暂停，从而将整体网络PFC暂停时间缩短一个数量级以上，这有助于避免PFC触发可能引起的性能问题。

2) 不同场景下的鲁棒性

在30%~90%的网络负载下，标准化FCT的第50百分位和第99百分位数如表3所示，值越低表示性能更优。与DCQCN、HPCC和BiCC相比，WRCC的平均FCT分别提升了24%~37%、8%~47%和10%~29%，尾FCT分别提升了45%~58%、47%~70%和10%~39%。随着网络负载的增加，WRCC的性能优势更加明显，符合设计思想。较高的网络负载会导致跨智算中心网络下更多突发流量与更长的队列，WRCC通过减少队列长度和PFC暂停时间来增强网络性能。



(a) 网络搜索模式流完成时间　(b) 数据挖掘模式流完成时间　(c) 谷歌RPC模式流完成时间

(d) 网络搜索模式队列长度累积分布　(e) 数据挖掘模式队列长度累积分布　(f) 谷歌RPC模式队列长度累积分布

图8　不同流量模式下的平均/尾FCT及交换机队列长度累积分布

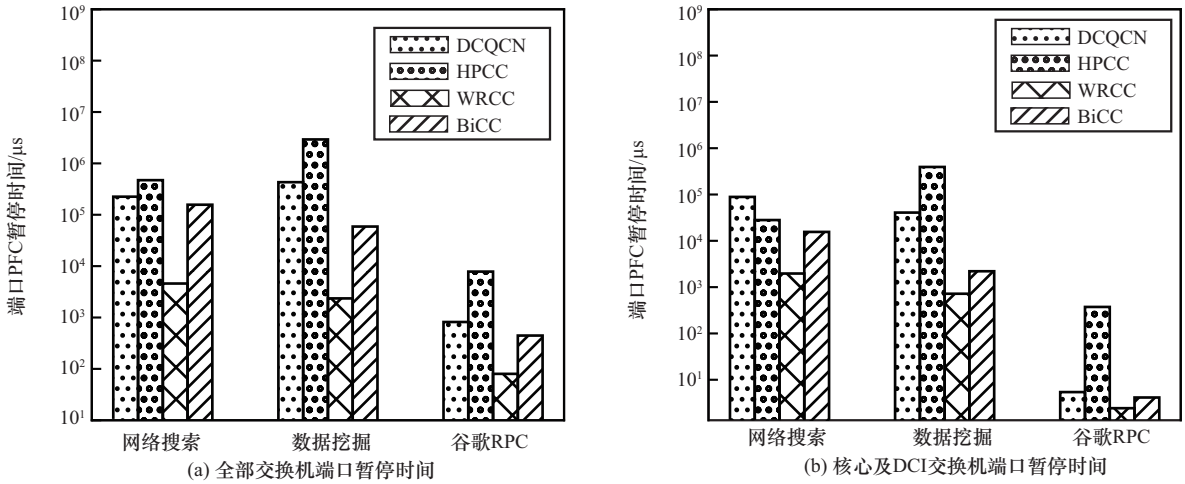


图9　不同流量模式下的交换机端口PFC暂停时间总和

表3　不同网络负载下WRCC的鲁棒性

网络负载	方法	50% FCT SlowDown	99% FCT SlowDown
30%	WRCC	**1.96**	**18.41**
	DCQCN	2.87	33.33
	HPCC	2.14	34.97
	BiCC	2.17	20.38
60%	WRCC	**4.12**	**25.46**
	DCQCN	6.56	51.16
	HPCC	7.74	83.81
	BiCC	5.84	28.9
90%	WRCC	**8.82**	**30.86**
	DCQCN	11.56	72.64
	HPCC	15.51	103.4
	BiCC	9.67	50.46

为了确定交换机采样周期对WRCC的性能影响，本文在不同的WRCCT值下进行了性能比较。选择最大智算中心内RTT的2倍、4倍和8倍作为交换机采样周期。不同WRCCT值下的实验结果如表4所示。结果表明，不同WRCCT之间的性能差异较小，且与其他方法相比，WRCC的性能均有所提升。

4.2　原型系统实验

4.2.1　原型系统实验设置

1) 算法实现与性能比较。本文在Tofino2 400 Gbit/s可编程交换机DCS810上实现了WRCC交换机算法。由于P4语言不支持除法操作，难以完整实现WRCC算法，本文通过移位操作实现近似除法，即将除数对齐至最近的2的幂次方，从而在硬件可编程性与算法功能之间取得折中。但该近似处理会引入一定的量化误差，该实验将实测环境下可配置参数尽量对齐至2的幂次方，以缓解此类量化误差。此外，本文还利用交换机的循环报文机制实现周期性定时操作。在终端侧，在400 Gbit/s Intel DK-DEV-AGI027的FPGA板卡上实现WRCC端侧算法，以及完成RoCEv2报文的INT头部添加和解析操作。在原型系统实验中，将WRCC算法与Mellanox 400 Gbit/s CX7网卡的DCQCN进行性能对比。

表4　不同关键参数下WRCC的鲁棒性

WRCCT/μs	方法	50% FCT SlowDown	99% FCT SlowDown
24	WRCC	**4.12**	**25.46**
	DCQCN	6.56	51.16
	HPCC	7.74	83.81
	BiCC	5.84	28.9
48	WRCC	**4.62**	**21.52**
	DCQCN	6.77	56.21
	HPCC	11.45	62.73
	BiCC	5.19	29.43
96	WRCC	**4.43**	**25.25**
	DCQCN	5.98	43.38
	HPCC	8.6	69.88
	BiCC	5.05	25.43

2) 实验设置。如图10所示，原型系统实验拓扑包含两个测试场景：跨域长距离测试场景与智算中心内短距离测试场景。测试场景包含数台用于640 km远距离光传输的设备、3台400 Gbit/s可编程交换机和3台服务器，每台服务器配置为双Intel Xeon Gold 5416S（16核，2 GHz）与256 GB内存，操作系统版本为Ubuntu 22.04，以及3块支持WRCC的自研400 Gbit/s RDMA FPGA网卡和3块英伟达400 Gbit/s CX7网卡，CX7驱动版本为Mellanox OFED 23.10。P4程序使用SDE 9.13.1编译，Quartus 24.3作为主要的FPGA开发套件。在跨域长距离测试场景中，服务器通过400 Gbit/s可编程交换机和光传输设备互联，两根长距离光纤长度为640 km，带宽为400 Gbit/s。由于现有交换机缓冲区大小有限，难以支持长距离PFC，因此该场景为有损网络。对于智算中心内短距离测试场景，3台服务器通过可编程交换机互联，交换机开启了ECN功能，交换机与网卡配置均为默认推荐配置。

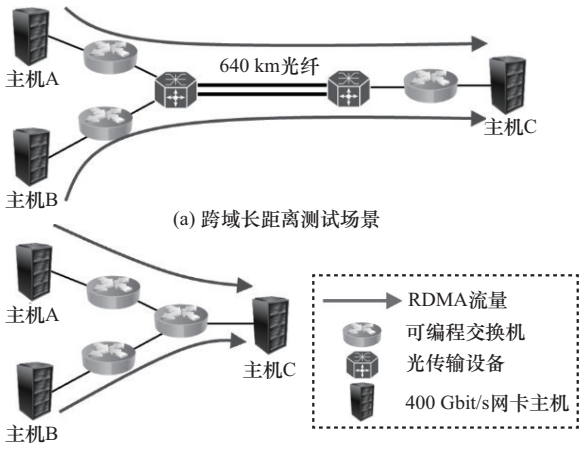


图10　原型系统实验拓扑与测试流量模式

3) 测试方法。本文采用RDMA基准测试工具Perftest[22]生成RDMA流，包括带宽流与时延流，以评估DCQCN与WRCC的实际性能。具体来说，通过构造Incast2:1拥塞的带宽流量作为背景流量，同时启动时延流量。默认带宽流量的MTU大小为1 024 B，测试流数目为1，消息大小为1 MB，持续时间为10 s，时延流量大小为64 B。通过动态调整带宽流量的消息大小、并发测试流数目，测量带宽流量的平均吞吐量以及时延流量的平均时延与尾时延，以衡量不同算法实际性能。

4.2.2　原型系统实验结果

如图11与图12所示，在不同消息与MTU大小的RDMA流量下，WRCC的平均时延与尾时延的差异远小于DCQCN。具体来说，在智算中心内短距离测试场景下，相较于DCQCN，WRCC在平均时延上与其性能接近，而在尾时延上性能提升幅度达到7%~49%。而在跨域长距离测试场景下，WRCC的性能提升幅度更加明显，平均时延和尾时延分别降低2%~7%和45%~49%。

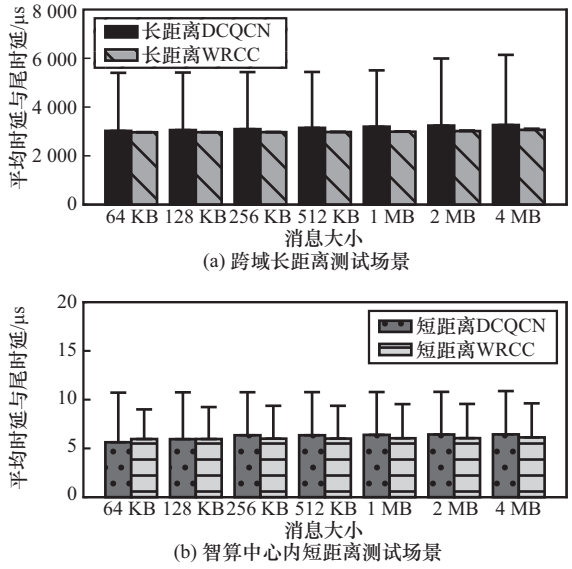


图11　不同消息大小下的平均时延与尾时延

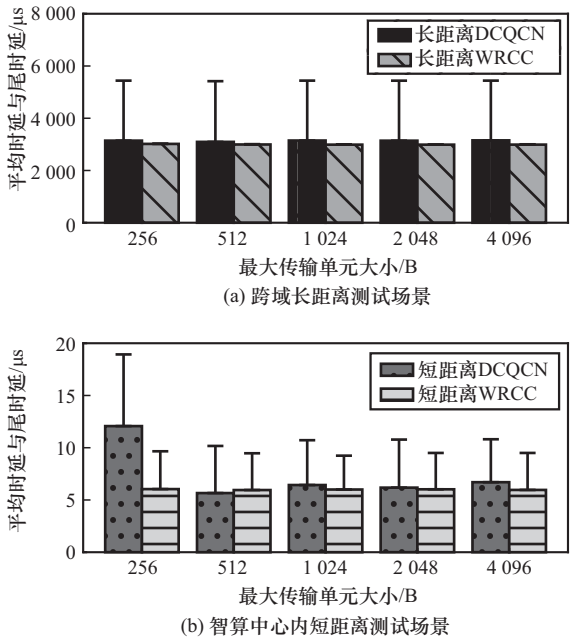


图12　不同最大传输单元下的平均时延与尾时延

本文通过动态调整带宽测试流量的持续时间，评估拥塞流量的平均吞吐量以衡量不同拥塞控制算法的收敛性。如图13所示，在跨域短距离RDMA测试场景下，由于智算中心内RTT较小，WRCC与DCQCN均能正常收敛，并实现公平带宽共享。然而，在跨域长距离RDMA测试场景下，WRCC的速率收敛速度显著高于DCQCN，在不同测试时间下均能实现更高的平均吞吐量，相较于DCQCN，WRCC能实现26%~90%的平均吞吐量提升。

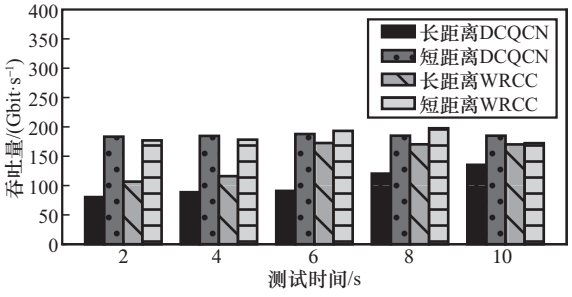


图13 不同测试时间下带宽流量的平均吞吐量

综上所述，在智算中心应用场景下，WRCC与DCQCN的吞吐量与平均时延接近，WRCC可实现7%~49%的尾时延提升。在跨域长距离有损场景下，WRCC能够实现更大的平均吞吐量和更低的时延。这种性能提升主要是因为基于交换机的公平速率计算策略能够实现更快的速率收敛。原型系统与大规模模拟在尾时延性能提升幅度上较为接近，在平均时延性能提升幅度上却存在差异。这是因为Perftest测试流量主要由持续长流构成，而仿真测试流量中突发短流的占比相对较高。DCQCN在长流控制方面具有较强的能力，但在应对突发短流时表现相对较弱。相比之下，基于交换机的速率限制策略在突发流量场景下能够实现更为显著的性能提升。

5 结束语

在算力网络应用场景中，传统拥塞控制方法难以应对长控制回路和混合流量拥塞带来的挑战，导致网络拥塞加剧及队列长度显著增加。本文提出了一种基于端网协同的跨智算中心网络拥塞控制方法WRCC。该方法能够动态计算每个拥塞端口的流量公平速率，由交换机为流量显式分配公平速率。这种显式分配机制在保持短队列和最小化PFC暂停触发的同时，实现了低时延和高吞吐量，有效缓解了混合流量拥塞下的速率收敛难题。长距离原型系统实验结果和大规模模拟实验结果表明，WRCC在不同流量模式下均表现出显著的性能提升，显著优于现有商用方法。未来工作将进一步聚焦于优化WRCC的交换机算法及参数配置，以适应更多样化的网络场景和流量模式。同时，考虑到算力网络拓扑的复杂性，还将探索如何将WRCC与负载均衡相结合，以实现高效的资源分配和动态流量调度。

参考文献：

[1] 刘韵洁, 汪硕, 黄韬, 等. 数算融合网络技术发展研究[J]. 中国工程科学, 2025, 27(1): 1-13.
Liu Y J, Wang S, Huang T, et al. Development of data and computing convergent network[J]. Strategic Study of Chinese Academy of Engineerng, 2025, 27(1): 1-13.

[2] Gangidi A, Miao R, Zheng S B, et al. RDMA over Ethernet for distributed training at meta scale[C]//Proceedings of the ACM SIGCOMM 2024 Conference. New York: ACM Press, 2024: 57-70.

[3] 王光全, 满祥锟, 徐博华, 等. 确定性光传输支撑广域长距算力互联[J]. 邮电设计技术, 2024(2): 7-13.
Wang G Q, Man X K, Xu B H, et al. Deterministic optical transmission for wide area and long-distance computing power interconnection[J]. Designing Techniques of Posts and Telecommunications, 2024(2): 7-13.

[4] Guo C X, Wu H T, Deng Z, et al. RDMA over commodity Ethernet at scale[C]//Proceedings of the 2016 ACM SIGCOMM Conference. New York: ACM Press, 2016: 202-215.

[5] Gao Y X, Li Q, Tang L B, et al. When cloud storage meets RDMA[C]//Proceedings of the 18th USENIX Symposium on Networked Systems Design and Implementation. Berkeley: USENIX Association, 2021: 519-533.

[6] Bai W, Abdeen S S, Agrawal A, et al. Empowering azure storage with RDMA[C]//Proceedings of the 20th USENIX Symposium on Networked Systems Design and Implementation. Berkeley: USENIX Association, 2023: 49-67.

[7] Singh A, Ong J, Agarwal A, et al. Jupiter rising: a decade of clos topologies and centralized control in google's datacenter network[J]. ACM SIGCOMM Computer Communication Review, 2015, 45(4): 183-197.

[8] Chen Y Q, Tian C, Dong J Q, et al. Swing: providing long-range lossless RDMA via PFC-relay[J]. IEEE Transactions on Parallel and Distributed Systems, 2023, 34(1): 63-75.

[9] Wan Z R, Zhang J, Yu M X, et al. BiCC: bilateral congestion control in cross-datacenter RDMA networks[C]//Proceedings of the IEEE INFOCOM 2024-IEEE Conference on Computer Communications. Pis-

cataway: IEEE Press, 2024: 1381-1390.

[10] Zhu Y B, Eran H, Firestone D, et al. Congestion control for large-scale RDMA deployments[J]. ACM SIGCOMM Computer Communication Review, 2015, 45(4): 523-536.

[11] Dukic V, Khanna G, Gkantsidis C, et al. Beyond the mega-data center: networking multi-data center regions[C]//Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication. New York: ACM Press, 2020: 765-781.

[12] Li Y L, Miao R, Liu H H, et al. HPCC: high precision congestion control[C]//Proceedings of the ACM Special Interest Group on Data Communication. New York: ACM Press, 2019: 44-58.

[13] Zeng G X, Bai W, Chen G, et al. Congestion control for cross-datacenter networks[J]. IEEE/ACM Transactions on Networking, 2022, 30(5): 2074-2089.

[14] Kim C, Sivaraman A, Katta N, et al. In-band network telemetry via programmable dataplanes[C]//Proceedings of the ACM SIGCOMM. New York: ACM Press, 2015: 1-2.

[15] Mittal R, Lam V T, Dukkipati N, et al. TIMELY: RTT-based congestion control for the datacenter[J]. ACM SIGCOMM Computer Communication Review, 2015, 45(4): 537-550.

[16] Kumar G, Dukkipati N, Jang K, et al. Swift: delay is simple and effective for congestion control in the datacenter[C]//Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication. New York: ACM Press, 2020: 514-528.

[17] Zhang Y R, Meng Q K, Hu C L, et al. Revisiting congestion control for lossless Ethernet[C]//Proceedings of the 21st USENIX Symposium on Networked Systems Design and Implementation. Berkeley: USENIX Association, 2024: 131-148.

[18] Taheri P, Menikkumbura D, Vanini E, et al. RoCC: robust congestion control for RDMA[C]//Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies. New York: ACM Press, 2020: 17-30.

[19] Zou S J, Huang J W, Liu J L, et al. GTCP: hybrid congestion control for cross-datacenter networks[C]//Proceedings of the 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS). Piscataway: IEEE Press, 2021: 932-942.

[20] Saeed A, Gupta V, Goyal P, et al. Annulus: a dual congestion control loop for datacenter and WAN traffic aggregates[C]//Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication. New York: ACM Press, 2020: 735-749.

[21] Long M F, Han J P, Wang W T, et al. LSCC: link-segmented congestion control for RDMA in cross-datacenter networks[C]//Proceedings of the 2024 IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS). Piscataway: IEEE Press, 2024: 1-10.

[22] Yu P W, Xue F Y, Tian C, et al. Bifrost: extending RoCE for long distance inter-DC links[C]//Proceedings of the 2023 IEEE 31st International Conference on Network Protocols (ICNP). Piscataway: IEEE Press, 2023: 1-12.

[23] Kachris C, Tomkos I. A survey on optical interconnects for data centers[J]. IEEE Communications Surveys and Tutorials, 2012, 14(4): 1021-1036.

[24] Filer M, Gaudette J, Yin Y W, et al. Low-margin optical networking at cloud scale[J]. Journal of Optical Communications and Networking, 2019, 11(10): C94.

[25] Zhong X L, Zhang J, Zhang Y L, et al. PACC: proactive and accurate congestion feedback for RDMA congestion control[C]//Proceedings of the IEEE INFOCOM 2022 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2022: 2228-2237.

[26] Borzunov A, Ryabinin M, Chumachenko A, et al. Distributed inference and fine-tuning of large language models over the Internet[C]//Proceedings of the 37th International Conference on Neural Information Processing Systems. New York: ACM Press, 2023: 12312-12331.

[27] Cai W B, Yang S L, Sun G, et al. Adaptive load balancing for parameter servers in distributed machine learning over heterogeneous networks[J]. ZTE Communications, 2023, 21(1): 72-80.

[28] Chen Z X, Shi L, Liu X D, et al. Boosting distributed machine learning training through loss-tolerant transmission protocol[C]//Proceedings of the 2023 IEEE/ACM 31st International Symposium on Quality of Service (IWQoS). Piscataway: IEEE Press, 2023: 1-10.

[29] Dukkipati N, Kobayashi M, Rui Z S, et al. Processor sharing flows in the Internet[C]//Quality of Service-IWQoS 2005. Berlin: Springer, 2005: 271-285.

[30] Balakrishnan H, Dukkipati N, Mckeown N, et al. Stability analysis of explicit congestion control protocols[J]. IEEE Communications Letters, 2007, 11(10): 823-825.

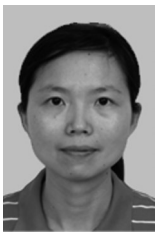
[31] Al-Fares M, Loukissas A, Vahdat A. A scalable, commodity data center network architecture[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(4): 63-74.

[32] Alizadeh M, Greenberg A, Maltz D A, et al. Data center TCP (DCTCP)[C]//Proceedings of the ACM SIGCOMM 2010 Conference. New York: ACM Press, 2010: 63-74.

[33] Joshi R, Song C H, Khooi X Z, et al. Masking corruption packet losses in datacenter networks with link-local retransmission[C]//Proceedings of the ACM SIGCOMM 2023 Conference. New York: ACM Press, 2023: 288-304.

[34] Greenberg A, Hamilton J R, Jain N, et al. VL2: a scalable and flexible data center network[C]//Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication. New York: ACM Press, 2009: 51-62.

[作者简介]



刘亚萍（1973– ），女，湖南常德人，博士，广州大学教授、博士生导师，主要研究方向为网络体系结构、智算中心网络、RDMA 技术、边缘计算、联邦学习、智能网络、网络安全等。

严定宇（1997– ），男，湖南常德人，北京邮电大学博士生，主要研究方向为数据中心网络传输优化、RDMA 协议优化等。

方滨兴（1960– ），男，江西万年人，博士，中国工程院院士，广州大学教授、博士生导师，主要研究方向为计算机体系结构、计算机网络、网络安全等。

许名广（1990– ），男，湖南邵阳人，鹏城实验室工程师，主要研究方向为数据中心网络传输、SDN、网络安全等。

张硕（1984– ），男，湖北枣阳人，博士，广州大学副教授、硕士生导师，主要研究方向为网络安全、数据中心网络、云计算、分布式系统等。

杨智凯（1996– ），男，山西晋城人，广州大学博士生，主要研究方向为网络安全、云计算、联邦学习等。