计 算 机 研 究 与 发 展
Journal of Computer Research and Development

https://crad.ict.ac.cn
62(5): 1290−1306, 2025

本期推荐

无丢失网络流量管理综述

张乙然[1,2]　王尚广[1,2]　任丰原[3]

[1]（北京邮电大学计算机学院（国家示范性软件学院）　北京　100876）

[2]（网络与交换技术全国重点实验室（北京邮电大学）　北京　100876）

[3]（清华大学计算机科学与技术系　北京　100084）

（yiranzhang@bupt.edu.cn）

Survey on Traffic Management in Lossless Networks

Zhang Yiran[1,2], Wang Shangguang[1,2], and Ren Fengyuan[3]

[1]（*School of Computer Science* (*National Pilot Software Engineering School*), *Beijing University of Posts and Telecommunications*, *Beijing* 100876）

[2]（*State Key Laboratory of Networking and Switching Technology* (*Beijing University of Posts and Telecommunications*), *Beijing* 100876）

[3]（*Department of Computer Science and Technology*, *Tsinghua University*, *Beijing* 100084）

Abstract　Lossless networks are increasingly widely used in high performance computing (HPC), data centers and other fields. Lossless networks use link layer flow control to ensure that packets will not be dropped by switches due to buffer overflow, thus avoiding loss retransmission and greatly improving the latency and throughput performance of applications. However, the negative effects introduced by link layer flow control (congestion spreading, deadlock, etc.) impose challenges for the large-scale deployment of lossless networks. Therefore, the introduction of traffic management technology to improve the scalability of lossless networks has received great attention. We systematically review the research progress of traffic management in typical lossless networks used in HPC and data centers including InfiniBand and lossless Ethernet. First, we introduce the negative impact of link layer flow control and the goals of traffic management, and summarize the traditional traffic management architecture of lossless networks. Then according to the traffic management technical route (congestion control, congestion isolation, load balancing etc.) and the driven location (sender-driven, receiver-driven, etc.), we classify and elaborate on the latest research progress of InfiniBand and lossless Ethernet traffic management, and analyze the corresponding advantages and limitations. Finally, we point out the issues that need to be explored in further research on lossless network traffic management, including unified architecture for traffic management, joint congestion management within the host and the network, and traffic management for domain applications.

Key words　lossless networks; InfiniBand; lossless Ethernet; traffic management; link layer flow control

摘　要　近年来，无丢失网络在高性能计算、数据中心等领域得到了广泛应用. 无丢失网络通过链路层流量控制技术保障网内交换机不会因缓存溢出而丢包，避免了数据丢失与重传，极大提高了应用的时延和吞吐量性能. 然而，链路层流量控制带来的负面效应（拥塞扩展、死锁等）使得无丢失网络的大规模部署面临着诸多挑战. 因此，引入流量管理技术来提升无丢失网络的可扩展性得到了更多关注. 对应用于高性能计算领域和数据中心领域的典型无丢失网络 InfiniBand 和无丢失以太网的流量管理研究进展进行系统

收稿日期：2024-02-21；**修回日期**：2024-12-18

基金项目：国家自然科学基金项目（62302055, 62132007, 62221003）；中央高校基本科研业务费专项资金

This work was supported by the National Natural Science Foundation of China (62302055, 62132007, 62221003) and the Fundamental Research Funds for the Central Unversities.

性综述，首先介绍链路层流量控制的负面影响和流量管理的目标，总结无丢失网络传统的流量管理架构. 然后根据流量管理技术路线（拥塞控制、拥塞隔离、多路径负载均衡等）以及驱动的位置（发送端驱动、接收端驱动等），对 InfiniBand 和无丢失以太网流量管理的最新研究进展进行分类和阐述，分析对应的优势以及局限性. 最后指出无丢失网络流量管理进一步研究中需要着重探索的问题，包括无丢失网络流量管理统一架构、主机内与网络联合流量管理以及面向领域应用的流量管理.

关键词 无丢失网络；InfiniBand；无丢失以太网；流量管理；链路层流量控制

中图法分类号 TP393

DOI: 10.7544/issn1000-1239.202440096 **CSTR:** 32373.14.issn1000-1239.202440096

近年来，更多具有严格要求的应用场景展现出对网络无丢失特性的需求. 例如，在大规模分布式存储业务中，丢包后的数据块重传会占据大量可用带宽，长时间超时可能导致传输几乎停止[1]. 在高性能计算（high performance computing, HPC）应用中，进程间同步通信流量的丢包甚至会导致系统崩溃[2]. 在现代数据中心中，数据包丢失导致的超时等待时间会极大延长时延敏感业务的完成时间[3-4]. 此外，随着算力效率的提升和大模型等新型算力应用的兴起，新型算力应用对网络时延、吞吐量等有了更高要求. 研究表明，在大模型场景中，1‰的网络丢包率将使模型有效计算时间占比下降 13%，极大影响模型训练效率[5]. 与此同时，远程访问内存（remote direct memory access, RDMA）技术由于其硬件卸载带来的服务器端时延和吞吐量的性能优势，逐渐成为各类新型算力应用的关键技术之一. 然而，RDMA 技术需要底层网络无数据包丢失才能发挥其最大性能优势[6].

因此，无丢失网络逐渐成为重要的网络基础设施. 无丢失网络是指在正常情况下数据包不会因网络中交换机缓存溢出而被丢弃的网络. 目前 2 类主流的无丢失网络是无丢失以太网和无限带宽网 InfiniBand. InfiniBand 发展于高性能计算领域，用于高性能计算机之间或者存储系统之间互联，支撑大规模专用计算任务的数据通信. InfiniBand 原生支持 RDMA 技术、基于虚拟通道的网络服务质量保证和以子网为单位的隔离功能. 无丢失以太网起源于基于以太网的数据中心，以面向用户服务为主，承载在线密集型、分布式存储、分布式计算等多种类型的应用. 随着数据中心应用对端到端时延、吞吐量等提出了更高的要求，RDMA 技术被引入到以传统 TCP/IP 协议为主的数据中心服务器，并在近年来发展出了基于融合以太网的 RDMA 传输协议 RoCE（RDMA over converged Ethernet）.

无丢失以太网和 InfiniBand 均使用链路层的流量控制机制来保证网络的无丢失特性，但其也附带了一些负面问题（如队头阻塞、死锁等），这影响了无丢失网络在带宽、时延、可靠性等方面的性能，因此，如何有效降低链路层流量控制的负面影响成为无丢失网络实现高效传输的关键挑战. 近年来，学术界和工业界设计提出了大量的流量管理方案，通过合理的网络资源分配来缓解或消除链路层流量控制的负面影响，保证数据包无丢失的同时提升应用的整体传输性能，从而推动无丢失网络的规模化部署. 本文将综述这 2 类无丢失网络在流量管理方面的最新进展.

本文对无丢失网络流量管理研究进行综述，主要贡献有 3 个方面：

1）以无丢失以太网和 InfiniBand 这 2 种典型的无丢失网络为例，介绍各自发展背景和应用特征，并从保障无丢失特性的链路层流量控制机制出发总结了无丢失网络流量管理的目标，归纳了无丢失以太网和 InfiniBand 流量管理的传统架构.

2）根据对流量管理驱动的位置或者技术手段，分别对 InfiniBand 和无丢失以太网的流量管理的最新研究进展进行分类，包括发送端驱动的拥塞控制、接收端驱动的拥塞控制、拥塞隔离、多路径负载均衡等，并分析总结对应的优势以及局限性.

3）对无丢失网络的流量管理研究进行了展望，指出了研究人员未来需要着重对无丢失网络流量管理统一架构、面向领域应用的流量管理以及机内与网络联合流量管理 3 个方向进行深入探索，才能使得无丢失网络具有更好的可扩展性和适用性.

1 背景介绍

1.1 无丢失网络分类

1）InfiniBand

在高性能计算中，InfiniBand 受到广泛应用. 据报道，在全球 TOP 500 榜单上的超算系统中，InfiniBand

占主导地位[7]; 高校、研究机构和政府机构的高性能计算集群中有 70% 部署了 InfiniBand[8].

为了避免因通信流量丢失而降低系统可靠性和损害延时性能, InfiniBand 行业协会(InfiniBand Trade Association, IBTA)在 InfiniBand 标准中就提出底层网络支持无丢失的特性, 采用基于信用的链路层流量控制(credit based flow control, CBFC)[9]. CBFC 工作在交换机端口的虚拟通道(virtual lane, VL)级别. VL 是共享一条物理链接的相互分立的逻辑通信链路, 每条物理链接可支持多达 15 条的标准 VL 和一条管理 VL. 如图 1(a)所示, 在 CBFC 中, 下游交换机为每个 VL 维护一个调整块寄存器(adjust block register, ABR)来记录接收到的块总数. 此外, 下游交换机周期性地向上游交换机发送流量控制信用限制(flow control credit limit, FCCL)消息, 消息内容是分配的缓冲区大小和 ABR 大小之和. 上游交换机为每个 VL 维护一个流量控制总发送块(flow control total block sent, FCTBS)寄存器来记录已发送的块总数. 收到 FCCL 消息后, 可用信用的数量即是 FCCL 和 FCTBS 数量之间数量的差值, 上游交换机只有在有可用信用时才能发送数据包, 每发送一个数据包, FCTBS 寄存器增加相应值, 相当于可用信用数量减少相应值.

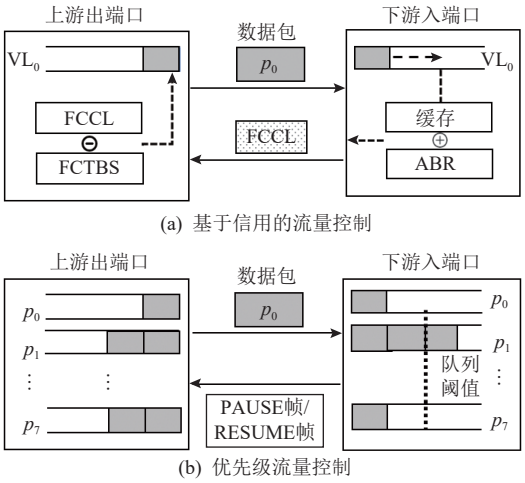


Fig. 1 Link layer flow controls in lossless networks

图 1 无丢失网络链路层流量控制

InfiniBand 与数据中心中的以太网具有不同的物理层和链路层. 早期 InfiniBand 的链路速度发展快于以太网. InfiniBand 交换机采用直通式转发以降低转发时延. 端主机侧原生支持旁路内核的远程访问内存 RDMA 技术, 大幅度提升了端主机的性能. 作为专用的网络技术, InfiniBand 需要特殊的网卡通道适配器(channel adapter, CA)和交换机, 支持以子网(subnet)为单位管理控制, 具有更高的建设成本和运营管理的复杂性.

InfiniBand 自诞生起便支撑高性能计算应用. 这类应用面向复杂的大规模科学与工程计算、数值模拟和数据分析等问题, 使用多个高性能计算机运行并行计算模型. 由于并行计算模型通常按照一定规则进行分割以实现并行数据或者并行任务, 因此从网络的角度看, 高性能计算应用的流量具有一定的模式和规律性. 具体而言, 高性能计算机广泛使用消息传递接口(message passing interface, MPI)进行进程间通信, MPI 支持 AllReduce, AlltoAll, Broadcast 等多种集合通信原语, 这类集合通信涉及多个发送方和接收方以实现结点间高效数据通信, 例如 AllReudce 通信原语将不同机器中的数据整合(reduce)之后再把结果分发给各个机器, Broadcast 通信原语将本地的数据分发给其他各个机器等.

2) 无丢失以太网

以太网凭借成本低、操作简单和兼容性广的优点, 一直是局域网的首选网络技术. 然而, 以太网这种保持简单、低成本的理念也导致需要为一些具有更严格要求的应用单独设计专用网络. 例如在多种业务需求下, 通信基础设施就需要包括多个不相交的网络, 如使用以太网技术的局域网、使用 Fibre Channel 技术的存储区域网和使用 InfiniBand 的高性能计算网络等. 维护这些专用网络带来了较高的成本, 同时增加了维护和管理的复杂性.

但是, 现代数据中心已普遍采用了在传统互联网中占主导地位的以太网. 鉴于此, IEEE 数据中心桥接(data center bridging, DCB)工作组致力于增强以太网的能力使之成为一个统一的交换结构, 从而取代传统的为特殊需求设计的各个独立的专用网络, 包括存储区域网、高性能计算网络和数据中心局域网等. 与维护多个独立的专用网络相比, 聚合增强型以太网(converged enhanced Ethernet, CEE)具有显著的性能、成本和管理优势. CEE 在链路层开发了逐跳的优先级流量控制(priority flow control, PFC)[10] 来支持多种应用场景需要的无丢失特性. 因此, CEE 也被称作无丢失以太网. 在无丢失以太网中, PFC 工作在交换机端口的队列级别, 如图 1(b)所示, 交换机入端口队列设置阈值 X_{on} 和 X_{off}, 当队长超过 X_{off} 时, 向上游发送暂停帧 PAUSE, 通知上游端口停止发送数据, 在队列长度降至 X_{on} 之后发送恢复帧 RESUME, 通知上游端口恢复数据包的发送, 最终确保了交换机不会因缓存溢出而丢弃数据包.

另一方面，RDMA 在数据中心的应用也促进了无丢失以太网的发展. 使用 RDMA 技术，数据传输不需要内核的参与，传输协议栈卸载到网卡上，数据直接从一个主机系统的存储区快速传输到另外一台远程主机的存储区. 与基于软件的 TCP/IP 网络协议栈相比，RDMA 硬件卸载和旁路内核的特性显著降低了 CPU 开销和整体时延. 例如，使用 RDMA 后，数据中心分布式机器学习的训练加快了 100 多倍[11]. 但根据经验，RDMA 的数据传输只有在没有数据包丢失时才能达到其最佳性能[6]. 因此，近年来基于融合以太网的 RDMA 传输协议 RoCE 以及扩展的支持 3 层 IP 网络的 RoCEv2[12] 得到了很大关注.

无丢失以太网近年来开始支撑数据中心的多种业务. 然而数据中心业务的通信模式具有几个显著的特征，对无丢失以太网的传输性能提出了更高的要求. 首先，数据中心业务往往具有高同步与高突发的特征. 以典型在线搜索应用的分割聚合模式为例，当底层大量机器被同时分配任务后，可能会在很相近的时间内同步完成并返回结果，造成高同步的并发流量通信模式 Incast[13]. 高同步的并发流可以在一个 RTT 内占满汇聚层机器的下行带宽，造成相应交换机网口缓存耗尽和 PFC 触发，进而导致网络中更多的交换机产生队列累积，使得网络通信的尾部时延增加. 与此同时，数据中心流大小通常表现出长尾分布[14]. 比如，在网页搜索业务负载[15] 中，50% 的流是小于 100 KB 的小流，而 80% 的数据量来自占比 10% 的大于 10 MB 的流. 随着数据中心链路带宽的增加，时延带宽积（bandwidth delay product, BDP）不断增大，网络中将存在很多在 1 个 RTT 内即可完成的突发性的小流，而端到端的拥塞控制很难及时控制小流，当很多小流并发到达时，极易造成短时间内交换机缓存耗尽和 PFC 的触发.

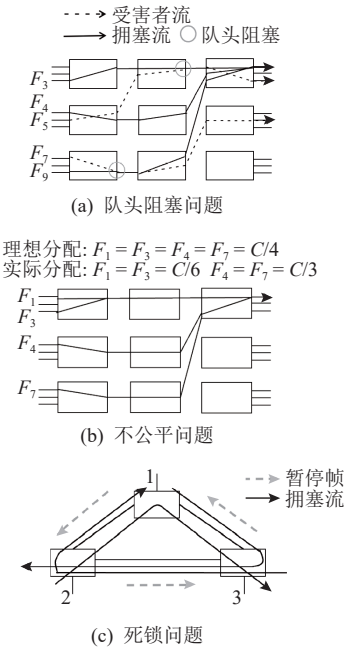
除此之外，iWARP[16] 是一种基于 TCP/IP 协议栈的远程直接内存访问技术，设计用于在广域网和局域网环境中提供高效、低延迟的数据传输. 与 InfiniBand 和 RoCE 相比，iWARP 直接构建于标准的以太网和 TCP/IP 协议之上，可以集成到现有的 IP 网络基础设施中，无需对网络架构进行大幅度改动. 由于 iWARP 基于 TCP/IP 协议栈，因此相比 InfiniBand 或 RoCE 可能引入更多的协议开销. 目前，iWARP 在一些特定领域中（需要与现有网络基础设施兼容或需要长距离、跨广域网连接等）具有较大应用前景.

1.2 无丢失网络流量管理的重要性

与传统有丢失网络不同，无丢失网络流量管理的目标有其特殊性，其来源于无丢失网络特有的链路层流量控制机制.

无丢失以太网和 InfiniBand 均使用链路层的流量控制机制来保证网络的无丢失特性，可以很好地避免瞬时拥塞导致的交换机丢包. 但是，链路层流量控制工作在交换机端口的队列或 VL 级别，一方面对于流是无感知的，另一方面队列数目和 VL 数目受到的交换机硬件限制是有限的，链路层流量控制如果被持续触发，可能带来一些附带损害.

1）队头阻塞问题. 如图 2（a）所示，当多条流量同时去往相同的目的地导致拥塞时，随着队列长度的增加，链路层流量控制向上游交换机触发，并逐渐向源端传播，网络中形成一棵拥塞树. 在拥塞流经过的路径上，流量即使不去往拥塞点也会被阻塞在队列中，带来了队头阻塞问题. 队头阻塞导致网络链路没有被充分利用，损害了无辜流量（也称受害者流）的性能. 队头阻塞问题往往与拥塞扩展同时出现，即拥塞点处的拥塞扩展到了网络中其他链路. 例如，在 InfiniBand 集群的测量研究表明，架顶交换机的上行链路端口在以 100 ms 为单位的观察间隔内，有约 60% 时间处于等待信用的状态[17].



(a) 队头阻塞问题

(b) 不公平问题

(c) 死锁问题

Fig. 2 Side effects of link layer flow control

图 2 链路层流量控制的负面影响

2）公平性问题. 如图 2（b）所示，4 条流去往同一目的地，理想情况下 4 条流应当均分瓶颈链路带宽，即各分得 $C/4$（假设链路带宽为 C）. 但是，由于链路层流量控制不能区分流，在交换机拥塞点处只能按

照不同入端口分配带宽, 最终导致共享同一端口队列或 VL 的 2 条流 F_1 和 F_3 得到了偏低的带宽分配, F_4 和 F_7 得到偏高的带宽分配.

3)死锁风险. 在实际部署中, 开启链路层流量控制可能使得网络有死锁的风险. 网络中可能由于路由配置错误、交换机固件故障或链路故障等多种原因导致网络中出现循环缓冲区依赖(cyclic buffer dependency, CBD), 一旦链路层流量控制触发时, 如图 2(c)所示, 由于 CBD 的存在, 在环路中的所有交换机都暂停了上游交换机端口队列或 VL 的发送, 并同时等待下游端口恢复数据包传输的信令, 网络会进入停滞状态, 极大降低传输性能[18-20].

1.3 无丢失网络流量管理传统架构

上述无丢失网络中链路层流量控制带来的负面影响, 其背后的根本原因是网络输入流量过量导致拥塞的发生, 而仅通过链路层流量控制, 无法实现合理的每流带宽分配. 因此, 有效的流量管理是必要的. 无丢失网络的流量管理的首要目标是对网络中流量进行合理控制, 消除网络中的拥塞, 避免链路层流量控制的持续触发, 缓解一系列负面影响, 是使能大规模无丢失网络的关键技术.

无丢失以太网和 InfiniBand 早期的相关标准中均制定了相似的流量管理架构, 以被动式流级别的拥塞控制为主要的技术路线, 主要包括拥塞探测、拥塞通知和拥塞控制 3 部分. 拥塞探测可以探测到网络中拥塞的出现, 用于检测网络中的拥塞情况, 同时进行拥塞识别以确定导致拥塞的拥塞流. 如果是网络内交换机负责实施拥塞探测与识别, 还需要拥塞通知将拥塞相关的信息通知给源端主机. 最终, 源端主机依靠端到端的拥塞控制对拥塞流进行相应的速率调整, 尽快解决网络中的拥塞. 其中, 拥塞探测是流量管理架构的基石, 它是流量管理中首先需要解决的问题, 它的探测结果决定了端到端拥塞控制的响应对象. 速率调节是端到端拥塞控制的核心, 它的调速结果决定了最终每流的带宽分配.

2 InfiniBand 流量管理研究进展

InfiniBand 建设成本高昂, 由于主要面向单一的高性能计算应用, 对于个性化的应用需求以及不同的物理集群通常可以定制不同的解决方案. 近年来, InfiniBand 流量管理发展出了多种技术路线, 除了传统流级别的拥塞控制外, 还有拥塞隔离、自适应路由以及端点主动拥塞控制等.

2.1 拥塞控制

1) 发送端驱动的拥塞控制

IBTA 制定的 InfiniBand 规范中的附录部分指定了 InfiniBand 网络中的拥塞控制框架和拥塞控制机制 IBCC[9]. 如图 3 所示, 交换机负责探测拥塞, 接收端主机负责将拥塞通知给发送端, CA 执行对注入流量的速率调节. 其中交换机依赖队长信息和信用值的有无来判断拥塞并识别拥塞流. 发送端的速率调节依赖预设的速率调节表 CCT. 发送端维护一个指针索引用于指向当前速率的表项, 通过定时器辅助的指针移动来改变当前速率, 从而实现增速或减速.

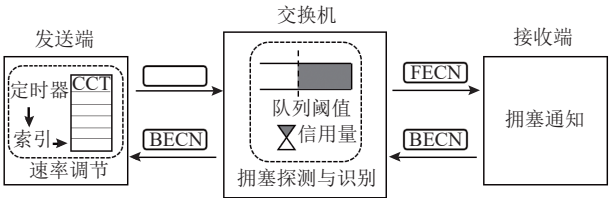


Fig. 3 Architecture of IBCC

图 3 IBCC 架构

挪威模拟研究实验室和迈络思的研究人员最早使用迈络思 ConnectX 系列网卡和 InfiniScale IV 交换机对 IBCC 进行相关实验, 结果表明 IBCC 经过合适的参数配置后可以有效缓解队头阻塞问题, 实现对典型高性能计算基准测试的吞吐量性能提升[21]. 但是, IBCC 维护速率调节表的方式一方面不能适应动态网络状况, 另一方面往往需要针对集群和工作负载特点进行大量参数调优才能获得理想的收敛性能. 因此, 尽管商用交换机已经支持了 IBCC, 但是其复杂的参数调优造成了其在实际中部署困难, 一些高性能计算集群甚至不开启 IBCC[17,22].

意识到 IBCC 参数调优的复杂性问题, 一些工作提出更快收敛、更易调优的 IB 拥塞控制算法. 核心思想在于对拥塞控制算法的减速部分提供更加明确的指导信息, 以减少逐步的启发式调节. 例如, Liu 等人[23] 提出利用链路利用率信息对 IBCC 进行增强; 最新的由 Zhang 等人[24] 提出的 RRCC, 采用接收端观察到的接收速率指导发送端的速率调节, 实现接收端驱动的速率调节和拥塞识别, 能够达到比 IBCC 更佳的收敛性能和对突发流量的容忍性.

2) 端点主动拥塞控制

针对高性能计算进程通信中存在的多对 1 流量模式易导致端点拥塞的问题, 同时为了克服被动式拥塞控制的根本性不足, 一些工作提出采用主动拥塞控制的思想来避免端点拥塞.

英伟达的研究人员在 SC 会议上提出 SRP[25]，该方案使用轻量级的预留握手协议提前避免拥塞. 发送端首先向接收端发送一个高优先级的预留数据包，指示其打算传输的数据量. 当接收端接收到该预留分组时，向发送端返回一个授权分组，指示可以发送的传输时间. 发送端在发出预留分组后直接发送投机数据包，而不必等待授权分组到达. 投机数据包在低优先级 VL 上传输，如果网络端点无拥塞，则投机数据包将完好无损地到达目的地. 然而，当网络端点出现拥塞时，投机数据包在网络中遇到较大的排队延迟，短暂超时后网络将丢弃投机数据包，并向发送端返回否定确认（negative acknowledgment, NACK）. 当发送端收到投机数据包的确认或 NACK 时则停止投机性传输. 当发送端达到指定的传输时间时，再以非投机模式传输消息的其余部分，并重新传输任何先前丢弃的数据包. 以非投机模式发送的数据包使用比投机数据包更高优先级的单独 VL，从而保证了投机数据包经历的任何排队延迟不会影响非投机数据包.

SRP 采用的预留协议会带来握手开销，当网络中短消息占主导时，握手开销对于消息性能可能是难以接受的. 因此，后续的研究工作 SMSRP[26] 提出不需要为每个消息进行提前预留操作，当消息准备好时，发送端立即以投机模式传输，消息使用低优先级 VL，并且在发生拥塞时可以被网络丢弃. 如果投机传输成功，也即当端点无拥塞时，SMSRP 几乎不会在网络中产生任何开销. 如果网络端点发生了拥塞，投机数据包可能被网络丢弃，网络交换机直接将相应的 NACK 发送回源发送端. 当源发送端收到 NACK 时，会向目的地发起预留请求以获取消息的重传时间. SRP 等机制由于只能解决最后一跳端点处发生的拥塞，因此提出与其他机制（如自适应路由等）配合使用来缓解网络内的拥塞.

2.2 拥塞隔离

针对链路层流量控制无法区分不同流而导致队头阻塞、不公平等问题的局限，InfiniBand 中一些研究工作提出了拥塞隔离的思想来从根本上解决链路层流量控制的附带损害. 在 InfiniBand 交换机中，基于信用的流量控制机制以 VL 为单位工作，属于不同 VL 的流量互不干扰. 与传统以太网交换机通常仅支持每端口最多 8 个队列不同，在 InfiniBand 交换机中，每个端口有 16 个 VL. 因此，可以使用不同 VL 来隔离拥塞流以避免阻塞其他流. 这类研究工作主要分为静态拥塞隔离和动态拥塞隔离.

1）静态拥塞隔离

静态拥塞隔离指根据提前获知的流量模式和网络拓扑结构定义流量类别到 VL 的映射. 挪威模拟研究实验室的研究人员提出了 vFtree[27]，根据 Fattree 网络拓扑路由时只需要决定流量上行链路的特点，提出将共享相同上游链路的流按照〈源发送端，目的接收端〉进行分组，并均匀分布在不同的 VL 上. 但是，随着网络规模越来越大，并且交换机支持的 VL 的数量有限（例如，每端口最多有 16 个 VL，通常 8 个 VL 可用于数据流量[28]），并非所有的流量都可以隔离在不同的 VL 中，仍然会有不同流量共享同一 VL. 尽管后续有研究人员提出了可以灵活调整 VL 映射的方案[29]，但总体而言静态拥塞隔离只能缓解部分的队头阻塞等问题.

2）动态拥塞隔离

动态拥塞隔离的主要思路是交换机根据动态的拥塞状况进行流量隔离，实现避免队头阻塞的同时高效利用交换机资源.

西班牙瓦伦西亚理工大学等的研究人员提出的 RECN[30-31] 是动态拥塞隔离的代表性工作. RECN 的核心思想是快速检测拥塞流并立即将其映射到单独的交换机队列. 通过允许到达交换机输入端口的非拥塞流的所有数据包共享同一队列，从而显著减少消除队头阻塞所需的队列数量. 拥塞流的所有数据包都将存储在单独的队列中，因此它们之间不会相互干扰，也不会干扰非拥塞流. 具体而言，当某个输入端口收到拥塞通知时，会在本地为去往拥塞输出端口的数据包分配预留队列（set aside queue, SAQ）. 所有请求拥塞的输出端口的数据包都将存储在该 SAQ 中. 如果拥塞持续存在，该 SAQ 的队长将持续增加. 当 SAQ 占用率达到阈值时，拥塞通知将传播到上游交换机. 上游交换机将维护状态信息，记录本端口为拥塞树叶子结点等. 随后，新的 SAQ 将被分配到该输出端口. 同样，当 SAQ 占用率达到阈值时，它会向将数据包转发到该输出端口的同一交换机上的所有输入端口通知拥塞，并用标志记录此类操作. 如果拥塞持续存在，则重复该过程，将拥塞通知和 SAQ 分配传播到受拥塞树影响的上游交换机，最终可能扩展到拥塞流的源发送端. SAQ 资源的释放与上述过程类似，当 SAQ 占用率为空或低于阈值时，释放该队列并通知下游结点，最终到达拥塞树的根结点.

交换机动态隔离方案能够近似模拟每条流使用单独的队列，很大程度上避免队头阻塞等问题，并且具有良好的可扩展性，能够支持较大规模的网络. 但

是，其需要较多的交换机资源实现一系列复杂的机制，尚未有硬件可部署的交换机解决方案.

2.3 多路径负载均衡

InfiniBand 网络中，去往同一目的结点往往有多条等价路径. 当网络内拥塞发生时，将发生拥塞的链路上的流量分散到其他链路利用率较低的等价路径上，可以暂时缓解网络内的拥塞. 因此一些研究工作提出使用先进的多路径负载均衡技术来缓解拥塞.

1)静态路由

静态路由通过预先计算的算法确定性地决定每一跳，保证网络内尽可能少地发生拥塞. 例如，西班牙瓦伦西亚理工大学的研究人员提出用于 Fattree 网络拓扑的确定性路由算法[32]，按照目的地址的顺序将其分配到不同的上行链路中，使得去往不同目的地址的流量会被路由到下一阶段的不同交换机. 这种针对连续目的地址的路由算法对于目的地址均匀分布的工作负载较友好，同时也不会引入数据包乱序问题. 实际上，静态路由算法通常只适用于特定的拓扑和流量模式，通常需要针对工作负载定制设计.

2)动态路由

动态路由根据网络拥塞情况确定每包走的路径. 主要分为基于源端的方案、基于交换机方案和基于集中控制器的方案.

早期美国 Myrinet 高性能计算集群使用了基于源端的分散路由方案[33]. 首先，当源端网卡因受到链路层流控影响而停止发送时，或者当被阻塞的数据包超过一个阈值后，认为网络中有拥塞发生，为数据包以一定概率随机选择其他的路径. 其次，为了避免重新选择的路径仍存在拥塞，在选择路径前，先发送微秒级探测包探测可选路径是否是拥塞的，重复此过程，直到最终找到可用路径或多次探测所有路由. 由于 Myrinet 中网络协议在设计时就对数据包乱序具有良好容忍性，因此分散路由引入的数据包乱序不会带来性能影响. 但是这种基于源端的分散路由方案通常只有端点视角的局部信息，无法保证快速将数据包分配到负载低的路径.

迈络思提出交换机直接根据端口负载状况选择等价路径，即交换机自适应路由(adaptive routing, AR)[34]. 交换机为流量动态选择拥塞程度最低的端口进行传输，并且支持调整路由的粒度. 拥塞程度基于出口队列负载状况(队长、链路利用率等)进行评估，可能需要针对不同拓扑和负载进行参数调整. 自适应路由使得相同流的不同数据包可能经过网络的不同路径，因此会出现数据包乱序问题. 由于 RDMA 传输协议默认数据按序到达，乱序数据包会被视作丢包从而重传，降低了 RDMA 传输的吞吐量. 因此，交换机自适应路由方案还需要端侧 RDMA 网卡的配合. 目前，一些商用网卡开始支持数据包的乱序交付，如英伟达的 ConnectX-7[35], BuleField-3 DPU[36] 等.

为了更好地适应工作负载的动态流量需求，美国劳伦斯利弗莫尔国家实验室等的研究人员提出了流感知的自适应路由 AFAR[37]. 控制器根据运行时更新的工作负载流量矩阵以及路由表，重复查找网络中最拥挤的链路，根据通过该链路接收数据的目标结点信息，为发送到该目标结点的所有数据重新选择路径以缓解网络内的拥塞. AFAR 需要对 MPI 库以及 InfiniBand 系统软件进行修改以实时获取任务负载信息和及时更新路由.

总体而言，动态路由的局限在于其不能解决拥塞的根源，即不能控制发送过量的发送端，只能短暂地缓解网络内部的拥塞，对于典型 Incast 流量模式导致网络最后一跳处的拥塞(也称端点拥塞)也无能为力.

2.4 其 他

除了上述 4 类流量管理方案，西班牙卡斯蒂利亚-拉曼查大学的研究人员指出发送端驱动的拥塞控制扩展性好，但反应时间长、收敛慢，而拥塞隔离能够迅速隔离拥塞流，但由于交换机支持队列数量有限，可扩展性差，因此提出了 CCFIT[38]，结合这 2 类方案的优势，相互补充，有效解决队头阻塞等问题. 一方面，动态拥塞流隔离可以在端侧尚未感知到拥塞的出现时消除部分队头阻塞；另一方面，对拥塞流的速率调节也会进一步降低在端口中同时存在许多拥塞流的可能性，减少所需物理队列数量. 实验表明 CCFIT 实现比 IBCC 更好的吞吐量性能，同时比动态拥塞隔离消耗更少的交换机资源.

2.5 InfiniBand 流量管理小结

InfiniBand 中被动拥塞控制以 IBCC 为代表，能够解决端点拥塞和网络拥塞，交换机设计简单、可扩展性高，但是其性能易受参数影响. 对于拥塞隔离机制，一方面其性能受限于目前交换机可用 VL 的数目，另一方面理想的动态解决方案需要复杂的交换机硬件设计，难以适用于大规模负载和网络. 自适应路由和端点主动拥塞控制机制不能同时适用于处理网络内和端点的拥塞，但可以作为缓解拥塞的辅助手段. 现有做法通常根据高性能计算负载的特点(例如流量模式是否均匀、有无端点拥塞等)，选择合适的流量管理手段. 表 1 总结对比了已有 InfiniBand 流量管理方案.

Table 1 Features Comparison of Main Traffic Management in InfiniBand

表 1 InfiniBand 主要流量管理特点对比

流量管理方案	代表算法	处理端点拥塞	处理网内拥塞	可部署性
被动拥塞控制	IBCC[9]	是	是	商用网卡已支持，需参数调优
	RRCC[24]	是	是	需修改网卡
拥塞隔离	RECN[30]	否	是	交换机设计实现复杂，尚无商用交换机支持
多路径负载均衡	vFtree[27]	是	是	无数据包乱序，但仅适用于特定拓扑和负载
	AR[34]	否	是	有数据包乱序，部分商用交换机和网卡已支持
	AFAR[37]	否	是	无数据包乱序，需要修改MPI库和子网控制器
端点主动拥塞控制	SRP[25]	是	否	接收端网卡设计复杂
	SMSRP[26]	是	否	接收端网卡设计复杂
其他	CCFIT[38]	是	是	交换机设计实现复杂，尚无商用交换机支持

3 无丢失以太网流量管理研究进展

无丢失以太网流量管理的主要技术手段是流级别的拥塞控制，以微软亚洲研究院 2015 年提出的 DCQCN[39] 为标志性工作，通过调节每流的发送速率解决网络中的拥塞，同时避免 PFC 的持续触发，提高应用的时延和吞吐量性能. 此外，受到基于 TCP 的传统数据中心网络影响，无丢失以太网近年来也开始利用多路径负载均衡进一步缓解网内热点. 在本节中，我们主要介绍拥塞探测、拥塞控制、多路径负载均衡方面的研究进展.

3.1 拥塞探测

根据拥塞探测执行的位置或者实体，现有研究进展可以分为基于交换机的拥塞探测、基于端主机的拥塞探测以及交换机和端主机协作的拥塞探测.

1）基于交换机的拥塞探测

在无丢失以太网中，IEEE DCB 工作组在 IEEE 802.1 Qau 中规定了量化拥塞通知（quantized congestion notification, QCN）[40]. 在 QCN 中，拥塞点（congestion point, CP）采样队列大小来计算拥塞的度量 F_b. 假设采样时刻瞬时队列为 Q，期望队列长度为 Q_{eq}，则度量值 F_b 根据上次采样时刻的队长增量 $Q_\delta = Q - Q_{old}$，以及采样时刻瞬时队列和期望队列的差值 $Q_{off} = Q - Q_{eq}$ 的线性组合计算.

但是，由于现代数据中心交换机普遍支持的是基于队列长度探测拥塞并使用 ECN 标记，无丢失以太网中广泛采用的拥塞探测机制是基于队列长度的拥塞探测. CP 根据 RED 算法[41] 标记 ECN. RED 算法包含 K_{max}, K_{min}, P 三个关键参数.当队长超过 K_{max} 时，交换机以概率 1 标记数据包；当队长处于 K_{max} 与 K_{min} 之间时，以概率 P 标记数据包；最后，当队长低于 K_{min} 时，不对数据包标记. 单个比特的 ECN 标记表示该流经历拥塞，即为拥塞流.

上述思想均为基于交换机队长进行拥塞状况的判断. 然而，Zhang 等人[42] 指出，仅基于队长的拥塞探测在无丢失以太网中容易造成错误的探测结果. 当端口因下游链路层流量控制触发而关闭时，也有可能有队列累积，而并非发生了拥塞. 此时根据队长标记 ECN 可能导致误伤受害者流. 因此，对无丢失网络交换机端口状态空间进行了重新划分. 定义第 1 种状态为拥塞态，端口持续发送，队长有累积. 第 2 种状态为待定态，端口发送模式为间歇性 ON-OFF，队长可能有累积. 第 3 种状态是不拥塞态，端口持续发送，无队长累积.

基于上述 3 种状态，文献 [42] 提出了交换机三元拥塞探测（ternary congestion detection, TCD）机制，考虑端口因流量控制触发的 ON-OFF 状态特征进行状态探测，能够准确区分出拥塞发生的端口、受到链路层流量控制触发影响的端口以及不拥塞的端口，重用了 ECN 的 2 b 标记位传递三元状态信息. 其中，处于待定态的端口的真实流量输入速率可能由于上游链路层流量控制的调制作用而被掩盖. 图 4 展示了三元状态的转换条件，其中的关键是区分出持续 ON 和 ON-OFF 的发送模式. TCD 给出了 ON 周期上界的计算方式，交换机可以提前配置 ON 周期的上界来区分 2 种发送模式. TCD 是为无丢失网络量身定制的拥塞探测机制，但需要对网络交换机进行改动.

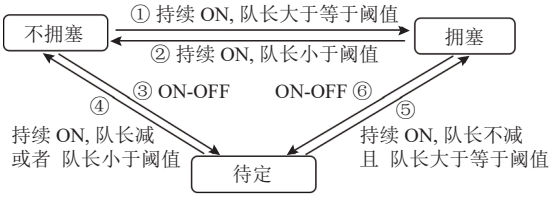


Fig. 4 Illustration of ternary states transitions in TCD

图 4 TCD 三态转换示意图

2）基于端主机的拥塞探测

隐式的丢包信号不再适用于无丢失网络. 因此，谷歌公司提出将基于 RTT 梯度的拥塞探测应用于无丢失以太网[43]. 基于 RTT 梯度的拥塞探测算法计算连续周期的 RTT 之间的差值，使用 EWMA 平滑来估计拥塞程度. 此外，文献 [44] 提出了类似的思想，在

高性能计算系统中使用分布式应用运行时间的梯度进行网络拥塞探测.

使用 RTT 或 RTT 梯度等端到端的延时信号, 无需网内设备支持, 具有灵活、可部署性高的优点. 但是, 由于无丢失网络中存在特有的链路层流量控制, 拥塞发生后, 一旦链路层流量控制向上游生效, 上游端口发生队头阻塞后, 即使是拥塞无关的流量也可能经历排队, 端主机会观察到延时的增加. 因此, 仅基于端到端的延时信息和端到端的拥塞探测方式可能难以区分因拥塞引起的延时增加和因逐跳流量控制生效引起的延时增加.

3)交换机和端主机协作的拥塞探测

最近无丢失以太网中的一些研究工作提出采用交换机和端主机协作的方式探测拥塞和识别拥塞流. Cheng 等人[45] 提出的 PCN 中给出了 NonPAUSE ECN (NP-ECN)的拥塞探测识别方法. 交换机维护一个计数器来记录被暂停数据包的数量. 只有未被暂停的数据包才允许使用 ECN 进行标记. 在接收端网卡处, 当被标记 ECN 的数据包的比例在一段时间内超过 1 个阈值(如 95%)时, 识别该流为拥塞流. NP-ECN 本质是一种启发式的方法, 认为拥塞流不会经历交换机端口暂停, 基于此将拥塞流筛选出来. NP-ECN 部分缓解了无丢失以太网中由于 PAUSE 帧导致的队长增加混淆拥塞探测的问题, 但是其周期性参数设置在不同网络拓扑下可能不适用.

Li 等人[11] 提出的 HPCC 通过已发送但还未确认的数据包数量判断拥塞程度. 这种方式需要依靠全网部署带内网络遥测(in-band network telemetry, INT)[46] 技术. 数据包传输时, 每经过一个交换机, 添加相应元信息字段来携带该交换机处传输的字节数和队列长度信息. 发送端根据路径上所有交换机的信息来最终计算最拥塞链路上正在传输但还未被确认的字节数. 总之, NPECN 和 INT 都不是交换机上独立的拥塞探测机制, 需要与端主机的拥塞控制算法协同配合才能探测拥塞以及识别拥塞流.

3.2 拥塞控制

根据拥塞控制驱动调速决策的位置或实体, 现有研究进展可以分为发送端驱动的拥塞控制、交换机驱动的拥塞控制和接收端驱动的拥塞控制.

1) 发送端驱动的拥塞控制

发送端驱动的拥塞控制的思想是先探测拥塞的出现, 发送端获取拥塞信号后做出速率调节. IEEE DCB 工作组提出的 QCN[40] 属于此类. 在其拥塞控制框架中, 使能 QCN 功能的网络交换设备 CP 负责拥塞探测, 探测到拥塞后, 交换机同时作为通知点(notification point, NP)产生显式拥塞通知信号直接发送给发送端, 最后发送端主机负责速率调节的反应点(reaction point, RP). 发送端采用乘性减速和类似 BICTCP[47] 的自增型二分速率增长算法. 其中分别使用了字节计数器和定时器根据发送数据量和发送时间自增加发送速率.

但是, QCN 基于 2 层网络设计, 不能在实际的 3 层 IP 网络中部署. 因此, 沿着 QCN 的思路, Zhu 等人[39] 在 2015 年基于商用交换机普遍支持的 ECN 功能提出了 DCQCN. 在 DCQCN 中, 接收端作为通知点, 将拥塞通知信号返回给发送端. 发送端如果在 1 个控制周期内收到拥塞通知信号, 则降低流的发送速率, 否则使用定时器和字节计数器决定流的增加速率, 速率增长规则与 QCN 类似. 受当时硬件条件限制, 控制周期采用 55 μs, DCQCN 根据 1 个控制周期内收到的拥塞信号, 采用启发式的逐步调节, 通过维护 ECN 比例的指数加权滑动平均值 α. 每当收到拥塞信号则乘性降 $\alpha/2$, 因此可能需要多个 RTT 才能收敛到理想速率. 目前商用 RoCEv2 网卡普遍支持 DCQCN, 是工业界广泛使用的无丢失以太网的拥塞控制算法.

莱斯大学和 IBM 的研究人员在 2014 年联合提出的 TCP Bolt[48], 是早期为无丢失以太网设计的流级别的拥塞控制, 其算法本质上是去掉了慢启动阶段的 DCTCP[15]. 但是基于拥塞窗口的 TCP Bolt 尚未在硬件网卡上得到支持, 而在终端主机协议栈上的实现会带来较大的 CPU 开销和延时.

几乎与 DCQCN 同时, Mittal 等人[43] 借鉴广域网中拥塞控制算法的思想, 为无丢失以太网提出了基于 RTT 梯度的 TIMELY 算法. TIMELY 指出现代网卡已经支持利用 ACK 对端到端时延进行微秒级测量, 因此 RTT 中蕴含了丰富的多比特拥塞信息, 可以在一定 RTT 范围内根据梯度启发式降低发送速率. 当 RTT 小于阈值 T_{low} 时, 认为网络无拥塞、队列空载, 可以增加发送速率; 当 RTT 大于阈值 T_{high} 时, 认为网络拥塞导致的排队严重, 应当减小发送速率; 当 RTT 介于二者之间时, 将梯度乘以一个减速系数调整发送速率. 但是, 这种基于梯度的方法有研究分析指出其在高度动态网络下的收敛性能不佳[49]. 2020 年, Kumar 等人[50] 对 TIMELY 算法进行改进提出了 Swift, Swift 使用时延的绝对值而不是梯度进行速率调节, 通过细粒度时延测量, 解耦网络时延和主机时延, 并使用了 2 个窗口分别处理主机内拥塞和网络拥塞. 窗口演变均使用 AIMD 原则, 相比 TIMELY, Swift 能够

更高效地处理大规模 Incast, 具有更好的时延性能和鲁棒性.

除了沿用传统有丢失网络的思路, 2019 年 Liu 等人[11] 提出了基于精确链路负载信息的高精度拥塞控制方案 HPCC. HPCC 的重要见解是认为已有方案慢收敛、长队列、调参复杂等问题的根本原因是传统网络缺乏细粒度的网络负载信息. 因此, 提出根据精确的链路负载信息感知每流状态进行细粒度调速. 对链路上正在传输但还未确认的字节数 I_j 进行控制, 采用基于窗口调整的方式, 使每条流路径上每个链路的 I_j 略低于链路带宽 $B_j \times T$(例如 95%). 对于链路 j, 每个发送方采用乘性增、乘性减(MIMD)和加性增(AI)结合的思想调整发送窗口. HPCC 需要全网部署 INT 技术, 带来了较高的每包携带开销, 可部署性较低. HPCC 追求非常低的交换机队列长度以获得较好的延时性能, 但牺牲了长流的吞吐量. 此外, 每包携带 INT 信息也进一步损害了吞吐量性能.

与已有沿用传统有丢失网络方法或者利用先进测量技术不同, Zhang 等人[51] 提出的拥塞控制算法 ACC 则独辟蹊径, 提出充分利用无丢失以太网自身的特性来指导发送速率的控制. 受 TCD 挖掘 PFC 对状态探测影响的启发, ACC 指出链路层流量控制 PFC 不仅影响了交换机端口状态, 而且还在保证中间交换机结点不丢包的同时, 保证了端到端整条路径的无丢失. 因此, 无丢失以太网具有一个重要的属性: 数据包守恒, 即发送入网络中的数据包数量总等于网络中排出的数据包数量. 基于该特性, ACC 提出了 2 个重要的拥塞控制指导原则: 首先, 应当使用 ACK 驱动的方法来推断拥塞流的网络管道容量以及精确过量的数据包数目; 其次, 为根本缓解无丢失以太网的队头阻塞问题, 应当对拥塞流进行充分抑制来尽快清除交换机中累积的数据包, 受害者流则应当根据拥塞扩展程度动态调整速率. ACC 需要与 TCD 配合使用, 识别出流状态后, 对拥塞流实行一种 2 步的调速策略: 首先拥塞流进入源停止(source halt)状态等待过量的数据包排出, 然后将发送速率与由 ACK 到达速率推断出的管道容量进行匹配. 对于受害者流, 则根据 TCD 标记中待定态(undetermined experienced, UE)的持续时间超过一定时间阈值后, 才进行减速调整. ACC 引入的源停止状态以最快的速度抑制了队头阻塞及其负面影响, 同时能够实现较好的延时和吞吐量性能.

2) 交换机驱动的拥塞控制

Taheri 等人[52] 提出的 RoCC 是交换机驱动的拥塞控制机制. 由交换机负责做出调速决策, 发送端只负责执行限速. RoCC 利用了交换机的可编程能力, 使用一个 PI 控制器计算出每流的公平速率通知给源端, 源端直接按照公平速率发送. 其中公平速率 R 的计算主要基于 3 个变量: 当前队长(Q_{cur})、队长变化方向($Q_{\text{cur}} - Q_{\text{old}}$)和参考队长偏差($Q_{\text{ref}}$), 即

$$R \leftarrow R - \alpha \times (Q_{\text{cur}} - Q_{\text{ref}}) - \beta \times (Q_{\text{cur}} - Q_{\text{old}}).$$

RoCC 交换机维护当前流的数目 N, 并根据 N 的大小自动调整 PI 控制器的参数 α 和 β. 对于较大的公平率值(较小的 N), α 和 β 应较大, 以确保快速收敛而不失去稳定性. 相反, 对于较小的公平率值(较大的 N), α 和 β 应该很小. 实验表明 RoCC 能够维持稳定的队列长度, 但是与传统数据中心一些基于交换机的调速方案类似, 仍存在计算复杂性较高等局限.

3) 接收端驱动的拥塞控制

为克服启发式调速的不足, 接收端驱动的拥塞控制通常由接收端显式指导发送端的速率调节, 避免了被动式的反应, 通常具有更好的收敛性能.

接收端驱动的拥塞控制最早由传统数据中心有丢失的以太网提出, 并不是专为无丢失以太网设计的, 但是通常被认为可以实现接近零的排队时延和数据包丢失. 例如 ExpressPass[53], pHost[54], NDP[55], Homa[56] 等工作. 这类拥塞控制的主要思想是以“先请求再分配”方式运行的主动传输, 即显式分配瓶颈链路的带宽并主动防止拥塞[53]. 因此能够达到较好的延时和吞吐性能. 这些方案通常让每条流在启动后的第 1 个 RTT 内线速传输数据包, 后续的传输将显式分配带宽. 在 ExpressPass 中, 接收端收到新流数据包后, 通过控制反馈的控制报文(如信用等)的发送速度, 以及交换机采取对控制报文精细化的限速, 来分配后续 RTT 数据的可用带宽, 能够避免端点处以及网络内的拥塞. pHost, NDP, Homa 采取类似的做法, 但是假设拥塞仅发生在端点处且网络内部不是瓶颈, 因此仅能处理端点处的拥塞. 然而, 这类主动拥塞控制中第 1 个 RTT 的数据包由于线速发送也可能导致网络内拥塞的产生[57]. 接收端驱动的主动拥塞控制需要接收端能够在为大量发送端发送控制报文的同时保证线速的性能, 实现代价高, 但尚未有在数据中心部署的案例.

Cheng 等人[45] 提出的 PCN 使用接收端观察到的接收速率指导发送端的部分速率调节, 是专为无丢失以太网设计的拥塞控制算法. 其能够较快收敛, 避免了类似 DCQCN, TIMELY 等的启发式调节, 减少了 PFC 的触发. PCN 指出拥塞流的接受速率是拥塞流

能够通过网络瓶颈的最大速率. PCN 的接收端将周期内拥塞流的接收速率直接通过 CNP 捎带给发送端, 发送端将该接受速率作为新的发送速率. 但是在严重拥塞和队头阻塞情况下, 交换机中积压的数据包队列可能仍需要较长时间才能消减, 难以从根本上缓解队头阻塞问题. 此外, PCN 需要同时修改交换机和网卡, 其交换机和端主机耦合的拥塞探测机制在不同网络配置下可能存在不准确性和需要参数调优.

RCC[58] 指出传统数据中心中大多数接收端驱动的方法默认拥塞发生在最后一跳却无法有效应对网内拥塞. 因此, RCC 根据最后一跳利用率来区分网内拥塞和最后一跳拥塞, 并采用不同的处理方式. 对最后一跳拥塞, 根据链路带宽和流数量可以直接计算出每流的可用带宽; 对网内拥塞利用比例-积分-微分(proportion-integration-differentiation, PID)控制器进行基于延时绝对差值和延时梯度的迭代式调节. 与 RoCC 在交换机上使用 PI 控制器基于队长计算速率不同, RCC 基于 RTT 参数进行发送窗口的迭代计算无需网络内设备的特殊支持. RCC 可以对最后一跳拥塞在 1 个 RTT 内实现快速收敛, 但仍存在接收端网卡硬件设计复杂等问题.

3.3 多路径负载均衡

近年来, 数据中心无丢失以太网在早期数据中心网络负载均衡研究基础之上, 发展出了基于交换机的负载均衡方案和基于端侧的负载均衡方案.

1) 基于交换机的负载均衡

根据多路径负载均衡决策的粒度, 已有研究工作可以分为数据包、flowlet 等级别的负载均衡机制. 2017 年, Ghorbani 等人[59] 提出了 DRILL, DRILL 在每个交换机上根据本地队列占用情况和随机算法对每个数据包决策来分配负载, 能够实现细粒度、均衡的各端口队列长度, 更好地应对数据中心突发流量. 但是, DRILL 为传统 TCP/IP 协议流量设计, 需要端侧协议栈配合解决数据包乱序问题.

几乎与 DRILL 同一时期, Alizadeh 等人[60] 提出了 CONGA 和 LetFlow[61], 巧妙利用 TCP 窗口机制自然形成的子流单元 flowlet 作为负载均衡的决策粒度, 交换机通过检测流量之间的到达间隔来识别 flowlet, 每个 flowlet 的数据包走同一路径, 可以带来较少的数据包乱序. 其中, CONGA 专门针对叶脊型的网络拓扑结构设计, 在 2 层交换机之间交换链路信息, 架顶交换机为 flowlet 选择链路利用率低的路径; LetFlow 则是 CONGA 的进一步优化, 指出 flowlet 的大小具有一定的弹性, 可以通过对每个 flowlet 随机选路的方式应对网络中出现的拓扑不对称等情况, 相比 CONGA 具有更低的交换机开销.

与 TCP 流量不同, 无丢失以太网中 RDMA 流量将产生很少的 flowlet, 大幅减少了使用 flowlet 进行路径选择的机会. 但是, 逐包负载均衡带来的乱序将极大影响 RDMA 的吞吐量性能, 因此提出了 Conweave[62]. 源端 ToR 层可编程交换机负责持续监控路径延迟并在检测到拥塞时重路由, 目的端借助于 ToR 层可编程交换机队列的暂停/恢复功能, 在数据平面为重路由的流维护一个正常队列和一个乱序队列, 通过控制 2 个队列的开闭来实现数据包最终按顺序发送. 由于 Conweave 全部在网内交换机实现, 端侧 RDMA 网卡对 Conweave 的重路由和乱序重组操作无感知, 保证了 RDMA 吞吐量性能和良好的负载均衡效果.

2) 基于端侧的负载均衡

基于端侧的负载均衡方案以端侧主动多径调度和端侧被动重路由为主要思想.

2015 年, He 等人[63] 提出了 Presto, 将多路径负载平衡功能实现在网络边缘的虚拟交换层(hypervisor), 从而无需改动网络硬件设备. Presto 将流量划分成 64 KB 等大小的块(flowcell)并主动散射流量, 利用网络拓扑中的对称性来确保 1 对主机之间的所有路径上负载均匀, 但是其需要在虚拟交换层进行数据分块和重排, 以及利用网络集中控制器的调度解决拓扑不对称导致的负载分布偏差的问题.

2018 年, Lu 等人[64] 提出了专为无丢失以太网设计的 MP-RDMA, 借鉴 TCP 的多路传输协议 MPTCP[65] 的思想, 端侧 RDMA 网卡支持主动从多个路径发送数据包. 通过多路径 ACK 时钟机制, 为所有路径维护一个拥塞窗口, 基于逐包 ACK 捎带的路径信息, 以拥塞感知的方式分配流量, 而无需维护每个路径的状态. 通过乱序感知的路径选择机制, 控制乱序交付数据包的程度, 从而最小化与其相关的元数据, 进一步降低 RDMA 网卡内存开销. 但是, 目前商用 RDMA 网卡还未支持多路径传输.

Hu 等人[66] 提出了无丢失以太网中 PFC 感知的端侧被动路由方案 Proteus, 指出在部署 PFC 的无丢失以太网中, 链路利用率、端到端的 ECN/RTT、交换机局部队长等信号均不能够独自准确反映 PFC 暂停下的路径状态, 因此, 提出通过使用多级信号的组合(如链路负载、队列长度、RTT 和 ECN 等), 检测三元路径状态(拥塞、不拥塞、待定)以做出初始路由决策, 并利用亚 RTT 级信号(如交换机中累积停留时间)反映瞬时的 PFC 暂停, 当检测出网络中 PFC 触发时

及时重路由并尽量避免乱序. 总体而言, 端侧重路由的方法一般需要 RTT 级别的响应时间才能被动地调整决策, 不能提前避免网络内热点的发生.

3.4 无丢失以太网流量管理小结

无丢失以太网的流量管理除了需要解决链路层流量控制的负面问题, 还需要满足数据中心时延敏感业务的低时延需求以及吞吐敏感业务的吞吐量要求, 因此其技术发展受数据中心传统以太网影响很大. 例如, 无丢失以太网早期技术路线以可扩展性高的拥塞控制算法为主, 从沿用传统拥塞信号 ECN 和 RTT, 到利用先进遥测技术 INT, 然后近年来才发展为根据无丢失以太网的自身特性量身定制拥塞探测信号与拥塞控制算法等. 其中交换机驱动的拥塞控制和接收端驱动的主动拥塞控制部署代价较高, 需要对现有交换机和硬件网卡做复杂的修改, 较难适用于高速可扩展的无丢失以太网. 此外, 近年来随着网络带宽增加以及网络路径数量增多, 多路径负载均衡也开始受到重视, 并发展出了基于端侧和交换机的解决方案. 表 2 和表 3 分别总结对比了无丢失以太网拥塞控制算法和流量负载均衡相关工作.

Table 2 Features Comparison of Main Congestion Control Algorithms in Lossless Ethernet

表 2 无丢失以太网主要拥塞控制算法的特点对比

分类	算法	拥塞信号	收敛速度	可部署性
发送端驱动	QCN[40]	量化拥塞通知	较快	不支持 IP 路由的网络
	DCQCN[39]	ECN	较快	交换机配置 ECN
	TIMELY[43]	RTT	较快	无需修改交换机
	HPCC[11]	INT	快	全网部署 INT
	ACC[51]	TCD[42]	快	需要交换机支持 TCD[42], 修改网卡
交换机驱动	RoCC[52]		快	交换机设计复杂
接收端驱动	PCN[45]	NP-ECN	快	修改交换机、网卡
	RCC[58]	RTT	快	修改网卡

Table 3 Features Comparison of Main Load Balancing Schemes in Lossless Ethernet

表 3 无丢失以太网主要负载均衡方案的特点对比

分类	方案	粒度	乱序程度	功能	可部署性
基于交换机	DRILL[59]	数据包	高	需要网络控制器	需要修改交换机和软件协议栈
	CONGA[60]	Flowlet	低	支持非对称拓扑	部分商用交换机支持
	LetFlow[61]	Flowlet	低	支持非对称拓扑	部分商用交换机支持
	Conweave[62]	子流	无	支持非对称拓扑	仅需修改 ToR 层交换机
基于端侧	Presto[63]	Flowcell	高	需要网络控制器	需要修改软件协议栈
	MP-RDMA[64]	数据包	低	支持非对称拓扑	需要修改 RDMA 网卡
	Proteus[66]	数据包	低	支持非对称拓扑	需要修改 RDMA 网卡

4 趋势与展望

数据中心与高性能计算应用场景逐渐融合, 需要统一的流量管理架构. 无丢失以太网发展于数据中心, InfiniBand 发展并广泛应用于高性能计算, 二者本质上都属于高速互联网络. 但由于二者承载应用、发展历史等的不同, 形成了不同的流量管理技术路线. 但是, 近年来随着数据中心的规模化发展和深度学习等人工智能方向的兴起, 二者对底层基础设施的需求呈现重叠乃至融合的趋势. 例如, 数据中心逐渐大规模部署 GPU 等高性能计算设备以支撑搜索推荐、生成式 AI 等人工智能应用, 开始参考高性能计算领域普遍采用的 MPI 集合通信库等, 使用 NVIDIA 集合通信库(NVIDIA collective communications library, NCCL)[67]、DeepSpeed 框架[68] 等进行大规模并行训练并针对 GPU 进行优化以支持更大规模的人工智能模型和更加高效的并行加速. 近年来, 基于无丢失以太网的数据中心也开始探索与 InfiniBand 中类似的流量管理技术路线. 例如, Lu 等人[64] 提出的 MP-RDMA、Shalev 等人[69] 提出的 SRD 等技术都提出采用多路负载均衡的思想来暂时缓解网络内的拥塞; Goyal 等

人[70]提出的BFC尝试逐跳为每流分配队列并进行控制,解决队头阻塞以及端到端协议无法控制短流的问题;IEEE 802.1 Qcz[71]提出的在无丢失以太网中支持高性能计算、分布式存储以及数据中心应用的动态拥塞隔离方案;此外,由超微半导体、博通、微软等多家工业界公司联合发起的超以太网联盟(Ultra Ethernet Consortium, UEC)提出UEC传输协议[72],将综合多路径散射、拥塞控制等多种机制,特别为人工智能和高性能计算应用提供最佳性能的同时保留以太网的优势.在如今数据中心与高性能计算逐渐融合的趋势下,更需要对流量管理架构展开系统性研究,研究不同流量管理策略的适用性、交互性以及如何组合不同的流量管理策略,提出统一的流量管理方法.

1)面向机内拥塞和网内拥塞的联合流量管理.流量管理研究的传统观点是拥塞主要发生在网络结构内.然而,如今数据中心、高性能计算集群中链路带宽迅速增长,从40 Gbps, 100 Gbps发展为200 Gbps乃至400 Gbps等,与之相比,主机内PCIe总线带宽等资源发展速度却相对缓慢,这导致了主机内拥塞问题凸显,即支持NIC和CPU/内存之间数据交换的主机网络内出现拥塞.值得注意的是,主机内PCIe总线网络通常也是无丢失的(使用基于信用的流量控制机制),因此传统无丢失网络的边界扩展到了主机内部.不断增加的线路速率和存写密集型工作负载给每数据包的软件/硬件处理资源带来压力,导致CPU、DRAM带宽和PCIe出现瓶颈,从而在NIC和主机堆栈中形成队列,易发生端主机拥塞.来自康奈尔大学、谷歌等的研究人员在2023年SIGCOMM会议上提出了HostCC[73],该方案能收集主机拥塞信号(如PCIe互联的I/O控制器的缓存占用大小),捕获主机拥塞的精确时间和位置,引入亚RTT粒度的主机本地拥塞响应,在网络流量和主机本地流量之间分配主机资源,减少了主机上的排队,从而提高了应用程序吞吐量和尾部延迟性能.HostCC需要网卡支持高精度的时间戳功能以提供准确的延时测量,避免端主机处理时延抖动的干扰.HostCC是主要针对主机内拥塞的研究工作.但是,随着近年来主机内NVLink/NVSwitch[74]等新型互联技术的发展,主机内网络的交换结构呈现复杂化趋势,如何对主机内网络流量的瓶颈定位,如何应对主机内复杂拥塞、主机内亚RTT粒度的流量管理与已有针对网络内流量管理之间是否存在交互影响以及二者如何协同工作等,有待进一步深入研究.

2)面向领域应用的流量管理.研究表明,深度学习等人工智能方法的训练通信开销甚至占据了训练迭代时间的主要部分[75-78],并且具有典型的通信-计算迭代的周期性特征.考虑到这种应用特征,传统拥塞控制追求的公平性对于机器学习工作负载可能并不友好,不公平的拥塞控制反而可能加速机器学习工作负载[79-80].例如,通过调整传输协议的公平性参数或者网络链路感知的任务调度,使得同时共享链路的任务的通信和计算周期交错,从而最大化链路利用率,提高通信阶段的吞吐量性能.Katebzadeh等人[81]指出数据分析、图形处理等各种数据中心机器学习工作负载对网络带宽会表现出不同的敏感度:对于某些工作负载,即使可用带宽的小幅减少也会导致完成时间的显著增加;而对于其他工作负载,完成时间可能对可用带宽并不敏感,因此提出设计了一种应用程序感知的带宽分配框架Saba,根据应用程序的敏感性分配网络带宽,线下根据程序画像定量确定敏感度并与运行时带宽分配相结合,同时无需修改网络硬件.此外,一些工作提出使用流水线[75]或优先参数传输[82-84]的方法来重叠单个任务的计算和通信阶段以提高网络资源利用率和降低网络传输时间等.例如,TicTac[82]通过对底层计算图的关键路径分析来导出接近最优的参数传输时间表并强制执行网络传输顺序,从而允许计算和通信的最大重叠,并防止传统上由于参数传输的随机顺序而导致的掉队任务.总之,通过对领域应用特征系统性的认识和理解,设计面向领域应用的流量管理方案来高效提升应用性能具有很大前景,有待进一步研究.

3)光电混合无丢失网络的流量管理.近年来,构建光电混合网络成为提升大规模数据中心网络整体运算效能的重要趋势.谷歌公司提出了Jupiter组网架构[85],并在多年迭代发展中逐渐引入光交换技术,支持在网络中创建专用光通道连接数据中心内的不同部分,满足特定场景下大带宽和低延迟需求.Jupiter通过SDN协同光电2层优化混合网络组网,解决了高速互联扩容问题.微软公司也提出了数据中心Sirius组网架构[86],引入光交换以及流量切换控制、快速调度协议、拓扑和路由优化等技术.2021年,阿里巴巴公司提出一种基于快速光开关的数据中心光交换网络[87],通过采用标签控制和光数据流量控制算法,进一步降低网络时延和丢包率.然而,现有光电混合无丢失网络的流量管理仍面临诸多挑战,例如现有光电混合无丢失网络的路由算法大多存在循环缓冲区依赖、易加剧PFC导致的网络死锁问题.为

此，Zhao 等人[88]提出将光电混合网络转化为一个虚拟的层级网络，然后借助虚拟层级构造上下路由来消除死锁；此外，由于光交换的切换调度复杂，业务流量在线规划变得更加困难，因此上海交通大学的研究人员提出利用竞争比来分析不同物理互连架构性能的思路，设计鲁棒性拓扑优化方案[89-90]，使得一种光交换机配置方式能够同时应对多种流量方案. 目前，光电混合网络是无丢失网络的重要发展方向，但其流量管理需要结合光交换技术的特点对其带来的新挑战深入研究.

5 总　　结

为实现低时延、无数据丢失的网络传输需求，无丢失网络近年在高性能计算、数据中心等领域迅速发展. 但是，保证无丢失特性的链路层流量控制带来的负面影响(队头阻塞、死锁、不公平等)成为构建大规模无丢失网络的挑战. 因此，流量管理成为提升无丢失网络可扩展性的关键技术. 流量管理通过对流量进行合理的带宽分配等手段，避免链路层流量控制的负面影响，提升应用传输性能. 本文分别从 2 类无丢失网络流量管理的视角，按流量管理技术路线以及驱动的位置进行分类，分析总结对应的优势以及局限性，理解 2 类无丢失网络采用不同技术路线的原因. 最后讨论了无丢失网络流量管理的设计空间和发展趋势，为后续的研究奠定基础.

作者贡献声明：张乙然提出了论文思路并进行论文撰写；王尚广和任丰原提出意见并修改论文.

参　考　文　献

[1] Sherman B, Thordal M, Hanson K. NVMe over Fibre Channel [M]. Hoboken, NJ: John Wiley & Sons, 2019

[2] Aspencore Network. Congestion management clears a path through 10 GbE [EB/OL]. [2024-01-02]. https://www.edn.com/congestion-management-clears-a-path-through-10-gbe/

[3] Zhu Yibo, Kang Nanxi, Cao Jiaxin, et al. Packetlevel telemetry in large datacenter networks[C]//Proc of the 2015 ACM Conf on Special Interest Group on Data Communication. New York: ACM, 2015: 479−491

[4] Li Yuliang, Miao Rui, Kim C, et al. Lossradar: Fast detection of lost packets in data center networks[C]//Proc of the 12th Int Conf on Emerging Networking Experiments and Technologies. New York: ACM, 2016: 481–495

[5] The Research Institution of China Mobile. White paper on network evolution of intelligent computing center for AI large model[EB/OL]. 2023 [2024-01-02]. http://www.ecconsortium.org/Uploads/file/20230517/1684313521798632.pdf

[6] Guo Chuanxiong, Wu Haitao, Deng Zhong, et al. RDMA over commodity Ethernet at scale[C]//Proc of the 2016 ACM SIGCOMM Conf. New York: ACM, 2016: 202–215

[7] NVIDIA. InfiniBand accelerates six of the top ten supercomputers in the world, including the top three, and four of the top five on June's TOP500 [EB/OL]. [2024-01-02]. https://nvidianews.nvidia.com/news/infiniband-accelerates-six-of-the-top-ten-supercomputers-in-the-world-including-the-top-three-and-four-of-the-top-five-on-june-s-top500

[8] InfiniBand Trade Association. Life in the fast lane: InfiniBand continues to reign as HPC interconnect of choice [EB/OL]. [2024-01-02].https://www.infinibandta.org/lifeinthefastlaneinfinibandcontinues-toreignashpcinterconnectofchoice/

[9] InfiniBand Trade Association. InfiniBand architecture specification release 1.4 [EB/OL]. [2021-02-01]. https://cw.infinibandta.org/document/dl/8567

[10] IEEE. IEEE 802.1 Qbb-priority-based flow control [EB/OL]. [2024-01-02]. http://www.ieee802.org/1/pages/802.1bb.html.

[11] Li Yuliang, Miao Rui, Liu H H, et al. HPCC: High precision congestion control[C]//Proc of the ACM Special Interest Group on Data Communication. New York: ACM, 2019: 44−58

[12] InfiniBand Trade Association. Supplement to InfiniBand architecture specification volume 1 release 1.2. 1. annex a17: RoCEv2 [EB/OL]. [2020-12-01]. https://cw.infinibandta.org/document/dl/7781

[13] Chen Yanpei, Griffith R, Liu Junda, et al. Understanding TCP throughput collapse in datacenter networks[C]//Proc of the 1st ACM Workshop on Research on Enterprise Networking. New York: ACM, 2009: 73–82

[14] Zeng Gaoxiong, Hu Shuihai, Zhang Junxue, et al. Overview of Data Center Network Transport Protocols[J]. Journal of Computer Research and Developement, 2020, 57(1): 74−84 (in Chinese)
(曾高雄, 胡水海, 张骏雪, 等. 数据中心网络传输协议综述[J]. 计算机研究与发展, 2020, 57(1): 74−84)

[15] Alizadeh M, Greenberg A, Maltz D A, et al. Data center TCP (DCTCP)[C]//Proc of the ACM SIGCOMM Conf. New York: ACM, 2010: 63–74

[16] IETF. A Remote Direct Memory Access Protocol Specification (RFC 5040) [EB/OL]. [2024-05-21]. https://datatracker.ietf.org/doc/html/rfc5040

[17] Alali F, Mizero F, Veeraraghavan M, et al. A measurement study of congestion in an InfiniBand network[C/OL]//Proc of the 2017 Network Traffic Measurement and Analysis Conf (TMA). Piscataway, NJ: IEEE, 2017[2024-05-21]. https://ieeexplore.ieee.org/document/8002911

[18] Qian Kun, Cheng Wenxue, Zhang Tong, et al. Gentle flow control: Avoiding deadlock in lossless networks[C]//Proc of the ACM Special Interest Group on Data Communication. New York: ACM, 2019: 75–89

[19] Hu Shuihai, Zhu Yibo, Cheng Peng, et al. Tagger: Practical PFC deadlock prevention in data center networks[J]. IEEE/ACM

Transactions on Networking, 2019, 27(2): 889902

[20] Hu Shuihai, Zhu Yibo, Cheng Peng, et al. Deadlocks in datacenter networks: Why do they form, and how to avoid them[C]//Proc of the 15th ACM Workshop on Hot Topics in Networks. New York: ACM, 2016: 92–98

[21] Gran E G, Eimot M, Reinemo S A, et al. First experiences with congestion control in InfiniBand hardware[C/OL]//Proc of the IEEE Int Symp on Parallel Distributed Processing (IPDPS). Piscataway, NJ: IEEE, 2010 [2024-02-21]. https://doi.org/10.1109/IPDPS.2010.5470419

[22] Pfister G, Gusat M, Denzel W, et al. Solving hot spot contention using InfiniBand architecture congestion control[C/OL]//Proc of the High Performance Interconnects for Distributed Computing. Piscataway, NJ: IEEE, 2005 [2024-02-21]. https://www.researchgate.net/publication/242408366

[23] Liu Qian, Russell R D, Gran E G. Improvements to the InfiniBand congestion control mechanism[C]// Proc of the 24th IEEE Annual Symp on High-Performance Interconnects (HOTI). Piscataway, NJ: IEEE, 2016: 27−36

[24] Zhang Yiran, Qian Kun, Ren Fengyuan. Receiver-driven congestion control for InfiniBand[C/OL]//Proc of the 50th Int Conf on Parallel Processing (ICPP). New York: ACM, 2021 [2024-02-21]. https://doi.org/10.1145/3472456.3472466

[25] Jiang Nan, Becker D U, Michelogiannakis G, et al. Network congestion avoidance through speculative reservation[C/OL]// Proc of the IEEE Int Symp on High-Performance Computer Architecture. Piscataway, NJ: IEEE, 2012 [2024-02-21]. https://doi.org/10.1109/HPCA.2012.6169047

[26] Jiang Nan, Dennison L, Dally W J. Network endpoint congestion control for finegrained communication[C/OL]//Proc of the Int Conf for High Performance Computing, Networking, Storage and Analysis. New York: ACM, 2015[2024-02-21]. https://doi.org/10.1145/2807591.2807600

[27] Guay W L, Bogdanski B, Reinemo S A, et al. vFtree — A Fattree routing algorithm using virtual lanes to alleviate congestion[C]// Proc of the 2011 IEEE Int Parallel Distributed Processing Symp. Piscataway, NJ: IEEE, 2011: 197−208

[28] HPC Advisory Council. Understanding basic InfiniBand QoS [EB/OL]. [2024-01-02]. https://hpcadvisorycouncil.atlassian.net/wiki/spaces/HPCWORKS/pages/1178075141/Understanding+Basic+InfiniBand+QoS

[29] EscuderoSahuquillo J, Garcia P J, Quiles F J, et al. A new proposal to deal with congestion in InfiniBandbased fattrees[J]. Journal of Parallel and Distributed Computing, 2014, 74(1): 1802−1819

[30] Duato J, Johnson I, Flich J, et al. A new scalable and costeffective congestion management strategy for lossless multistage interconnection networks[C]// Proc of the 11th Int Symp on High-Performance Computer Architecture. Piscataway, NJ: IEEE, 2005: 108−119

[31] Garcia P, Quiles F, Flich J, et al. Efficient, scalable congestion management for interconnection networks[J]. IEEE Micro, 2006, 26(5): 52−66

[32] EscuderoSahuquillo J, Garcia P, Quiles F, et al. Cost-effective congestion management for interconnection networks using distributed deterministic routing[C]//Proc of the 16th IEEE Int Conf on Parallel and Distributed Systems. Piscataway, NJ: IEEE, 2010: 355−364

[33] Geoffray P, Hoefler T. Adaptive routing strategies for modern high performance networks [C]//Proc of the 16th IEEE Symp on High Performance Interconnects. Piscataway, NJ: IEEE, 2008: 165−172

[34] NVIDIA. How to configure adaptive routing and self healing networking [EB/OL]. [2024-01-02]. https://enterprise-support.nvidia.com/s/article/How-To-Configure-Adaptive-Routing-and-Self-Healing-Networking-New

[35] NVIDIA. NVIDIA ConnectX-7[EB/OL]. [2024-05-21]. https://resources.nvidia.com/en-us-accelerated-networking-resource-library/connectx-7-datasheet

[36] NVIDIA. NVIDIA BlueField networking platform[EB/OL]. [2024-05-21]. https://docs.nvidia.com/networking/display/bf3dpu/introduction

[37] Smith S A, Cromey C E, Lowenthal D K, et al. Mitigating interjob interference using adaptive flowaware routing[C]// Proc of the Int Conf for High Performance Computing, Networking, Storage and Analysis(SC18). Piscataway, NJ: IEEE, 2018: 346−360

[38] EscuderoSahuquillo J, Gran E G, Garcia P J, et al. Combining congested-flow isolation and injection throttling in HPC interconnection networks[C]//Proc of the 2011 Int Conf on Parallel Processing. New York: ACM, 2011: 662−672

[39] Zhu Yibo, Eran H, Firestone D, et al. Congestion control for largescale RDMA deployments[C]//Proc of the 2015 ACM Conf on Special Interest Group on Data Communication. New York: ACM, 2015: 523–536

[40] IEEE. 802.1 Qau—Congestion Notification [EB/OL]. 2010[2024-01-02]. http://www.ieee802.org/1/pages/802.1au.html

[41] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance[J]. IEEE/ACM Transactions on Networking, 1993, 1(4): 397−413

[42] Zhang Yiran, Liu Yifan, Meng Qingkai, et al. Congestion detection in lossless networks[C]//Proc of the 2021 ACM Conf on Special Interest Group on Data Communication. New York: ACM, 2021: 370–383

[43] Mittal R, Lam V T, Dukkipati N, et al. TIMELY: Rttbased congestion control for the datacenter [C]//Proc of the 2015 ACM Conf on Special Interest Group on Data Communication. New York: ACM, 2015: 537–550

[44] Patke A, Jha S, Qiu Haoran, et al. Delay sensitivity-driven congestion mitigation for HPC systems [C]//Proc of the ACM Int Conf on Supercomputing. New York: ACM, 2021: 342–353

[45] Cheng Wenxue, Qian Kun, Jiang Wanchun, et al. Rearchitecting congestion management in lossless Ethernet [C]//Proc of the 17th USENIX Symp on Networked Systems Design and Implementation (NSDI 20). Berkeley, CA: USENIX Association, 2020: 19−36

[46] Open Compute Project. Inband network telemetry in broadcom trident3[EB/OL]. [2024-01-02]. https://www.opencompute.org/files/INTInBandNetworkTelemetryAPowerfulAnalyticsFrameworkforyourDataCenterOCPFinal3.pdf

[47] Xu Lisong, Harfoush K, Rhee I. Binary increase congestion control (bic) for fast long-distance networks[C]// Proc of IEEE INFOCOM.

Piscataway, NJ: IEEE, 2004: 2514−2524

[48] Stephens B, Cox A L, Singla A, et al. Practical DCB for improved data center networks[C]//Proc of the IEEE Conf on Computer Communications. Piscataway, NJ: IEEE, 2014: 1824−1832

[49] ZhuYibo, Ghobadi M, Misra V, et al. ECN or delay: Lessons learnt from analysis of DCQCN and TIMELY[C]//Proc of the Conf on Emerging Network Experiment and Technology. New York: ACM, 2016: 313−327

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

[51] Zhang Yiran, Meng Qingkai, Hu Chaolei, et al. Revisiting congestion control for lossless Ethernet[C]// Proc of the 21st USENIX Symp on Networked Systems Design and Implementation (NSDI 24). Berkeley, CA: USENIX Association, 2024: 131−148

[52] Taheri P, Menikkumbura D, Vanini E, et al. RoCC: Robust congestion control for RDMA[C]// Proc of the 16th Int Conf on Emerging Networking Experiments and Technologies. New York: ACM, 2020: 17−30

[53] Cho I, Jang K, Han D. Credit-scheduled delay-bounded congestion control for datacenters [C]//Proc of the Conf of the ACM Special Interest Group on Data Communication. New York: ACM, 2017: 239−252

[54] Gao P, Narayan A, Kumar G, et al. pHost: Distributed nearoptimal datacenter transport over commodity network fabric[C/OL]//Proc of the 11th ACM Conf on Emerging Networking Experiments and Technologies. New York: ACM, 2015[2024-02-21]. https://doi.org/10.1145/2716281.2836086

[55] Handley M, Raiciu C, Agache A, et al. Rearchitecting datacenter networks and stacks for low latency and high performance[C]//Proc of the Conf of the ACM Special Interest Group on Data Communication. New York: ACM, 2017: 29−42

[56] Montazeri B, Li Y, Alizadeh M, et al. Homa: A receiver-driven low-latency transport protocol using network priorities[C]//Proc of the 2018 Conf of the ACM Special Interest Group on Data Communication. New York: ACM, 2018: 221–235

[57] Hu Shuihai, Bai Wei, Zeng Gaoxiong, et al. Aeolus: A building block for proactive transport in datacenters [C]//Proc of the Conf of the ACM Special Interest Group on Data Communication. New York: ACM, 2020: 422−434

[58] Zhang Jiao, Zhong Xiaolong, Wan Zirui, et al. RCC: Enabling receiver-driven RDMA congestion control with congestion divide-and-conquer in datacenter networks[J]. IEEE/ACM Transactions on Networking, 2023, 31(1): 103−117

[59] Ghorbani S, Yang Zibin, Godfrey P B, et al. DRILL: Micro load balancing for low-latency data center networks[C]//Proc of the Conf of the ACM Special Interest Group on Data Communication. New York: ACM, 2017: 225−238

[60] Alizadeh M, Edsall T, Dharmapurikar S, et al. CONGA: Distributed congestion-aware load balancing for datacenters[C]//Proc of the 2014 ACM Conf on SIGCOMM. New York: ACM, 2014: 503−514

[61] Vanini E, Pan Rong, Alizadeh M, et al. Let It Flow: Resilient asymmetric load balancing with flowlet switching[C]//Proc of the 14th USENIX Symp on Networked Systems Design and Implementation (NSDI 17). Berkeley, CA: USENIX Association, 2017: 407−420

[62] Song C, Khooi X, Joshi R, et al. Network load balancing with in-network reordering support for RDMA[C]//Proc of the ACM SIGCOMM 2023 Conf. New York: ACM, 2023: 816–831

[63] He Keqiang, Rozner E, Agarwal K, et al. Presto: Edge-based load balancing for fast datacenter networks[C]//Proc of the 2015 ACM Conf on Special Interest Group on Data Communication. New York: ACM, 2015: 465−478

[64] Lu Yuanwei, Chen Guo, Li Bojie, et al. Multi-path transport for RDMA in datacenters[C]//Proc of the 15th USENIX Symp on Networked Systems Design and Implementation (NSDI 18). Berkeley, CA: USENIX Association, 2018: 357−371

[65] Wischik D, Raiciu C, Greenhalgh A, et al. Design, implementation and evaluation of congestion control for multipath TCP[C/OL]// Proc of the 8th USENIX Symp on Networked Systems Design and Implementation (NSDI 11). Berkeley, CA: USENIX Association, 2011[2024-02-21]. http://www.usenix.org/events/nsdi11/tech/full_papers/Wischik.pdf

[66] Hu Jinbin, Zeng Chaoliang, Wang Zilong, et al. Enabling load balancing for lossless datacenters[C/OL]//Proc of the 31st IEEE Int Conf on Network Protocols (ICNP). Piscataway, NJ: IEEE, 2023 [2024-02-21]. https://doi.org/10.1109/ICNP59255.2023.10355615

[67] Microsoft. MSCCL[EB/OL]. [2024-01-02]. https://github.com/microsoft/msccl

[68] Microsoft. DeepSpeed[EB/OL]. [2024-01-02]. https://github.com/microsoft/DeepSpeed

[69] Shalev L, Ayoub H, Bshara N, et al. A cloud-optimized transport protocol for elastic and scalable HPC[J]. IEEE Micro, 2020, 40(6): 67−73

[70] Goyal P, Shah P, Zhao K, et al. Backpressure flow control[C]//Proc of the 19th USENIX Symp on Networked Systems Design and Implementation (NSDI 22). Berkeley, CA: USENIX Association, 2022: 779−805

[71] IEEE. IEEE 802.1 Qcz—Congestion Isolation [EB/OL]. 2019[2024-01-02]. https://1.ieee802.org/tsn/8021qcz/

[72] Ultra Ethernet Consortium. Ultra Ethernet consortium [EB/OL]. 2023[2024-01-02]. https://ultraethernet.org

[73] Saksham A, Arvind K, Rachit A, et al. Host congestion control [C]//Proc of the ACM SIGCOMM 2023 Conf. New York: ACM, 2023: 275−287

[74] NVIDIA. NVLink and NVSwitch: Fastest HPC fata center platform [EB/OL]. [2024-01-02]. https://www.nvidia.com/en-us/data-center/nvlink/

[75] Huang Yanping, Cheng Youlong, Bapna A, et al. GPipe: Efficient training of giant neural networks using pipeline parallelism[C]//Proc of the 33rd Int Conf on Neural Information Processing Systems. New York: ACM, 2019: 103−112

[76] Khani M, Ghobadi M, Alizadeh M, et al. SiP-ML: High-bandwidth

optical network interconnects for machine learning training[C]//Proc of the Conf of the ACM Special Interest Group on Data Communication. New York: ACM, 2021: 657−675

[77] Narayanan D, Harlap A, Phanishayee A, et al. PipeDream: Generalized pipeline parallelism for DNN training[C]//Proc of the 27th ACM Symp on Operating Systems Principles. New York: ACM, 2019: 1−15

[78] Wang Shuai, Li Dan. Research progress on network performance optimization of distributed machine learning system[J]. Chinese Journal of Computers, 2021, 45(7): 1384−1411 (in Chinese)
(王帅, 李丹. 分布式机器学习系统网络性能优化研究进展[J]. 计算机学报, 2021, 45(7): 1384−1411)

[79] Rajasekaran S, Ghobadi M, Kumar G, et al. Congestion control in machine learning clusters[C]//Proc of the 21st ACM Workshop on Hot Topics in Networks. New York: ACM, 2022: 235−242

[80] Rajasekaran S, Ghobadi M, Akella A. CASSINI: Network-aware job scheduling in machine learning clusters[C]//Proc of the 21st USENIX Symp on Networked Systems Design and Implementation (NSDI 24). Berkeley, CA: USENIX Association, 2024: 1403−1420

[81] Katebzadeh M, Costa P, Grot B. Saba: Rethinking datacenter network allocation from application's perspective[C]//Proc of the 18th European Conf on Computer Systems (EuroSys). New York: ACM, 2023: 623−638

[82] Hashemi S H, Abdu J, Campbell R. TicTac: Accelerating distributed deep learning with communication scheduling[C]//Proc of the 1st Machine Learning and Systems. California: MLSys, 2019: 418−430

[83] Jayarajan A, Wei J, Gibson G, et al. Priority-based parameter propagation for distributed DNN training[C]// Proc of the 1st Machine Learning and Systems. California: MLSys, 2019: 132−145

[84] Peng Yanghua, Zhu Yibo, Chen Yangrui, et al. A generic communication scheduler for distributed DNN training acceleration[C]//Proc of the 27th ACM Symp on Operating Systems Principles. New York: ACM, 2019: 16−29

[85] Poutievski L, Mashayekhi O, Ong J, et al. Jupiter evolving: Transforming Google's data center network via optical circuit switches and software-defined networking[C]//Proc of the ACM SIGCOMM 2022 Conf. New York: ACM, 2022: 66−85

[86] Ballani H, Costa P, Behrendt R, et al. Sirius: A flat datacenter network with nanosecond optical switching[C]//Proc of the ACM SIGCOMM 2020 Conf. New York: ACM, 2020: 782−797

[87] Xue Xuwei, Pan Bitao, Chen Sai, et al. Experimental assessments of fast optical switch and control system for data center networks [C/OL]//Proc of the 2021 Optical Fiber Communications Conf and Exhibition (OFC). Piscataway, NJ: IEEE, 2021[2024-02-21]. https://ieeexplore.ieee.org/document/9489828

[88] Zhao Shizhen, Zhang Qizhou, Cao Peirui, et al. Flattened clos: Designing high-performance deadlock-free expander data center networks using graph contraction[C]// Proc of the 20th USENIX Symp on Networked Systems Design and Implementation (NSDI 23). Berkeley, CA: USENIX Association, 2023: 663−683

[89] Zhao Shizhen, Cao Peirui, Wang Xinbing. Understanding the performance guarantee of physical topology design for optical circuit switched data centers[J]. Measurement and Analysis of Computing Systems, 2022, 5(3): 1−24

[90] Cao Peirui, Zhao Shizhen, Teh M Y, et al. TROD: Evolving from electrical data center to optical data center[C/OL]//Proc of the 29th IEEE Int Conf on Network Protocols (ICNP). Piscataway, NJ: IEEE, 2021[2024-02-21]. https://doi.org/10.1109/ICNP52444.2021.9651977



Zhang Yiran, born in 1995. PhD, associate professor, PhD supervisor. Member of CCF. Her main research interests include network traffic management and control, datacenter network, and satellite network.

张乙然, 1995 年生. 博士, 副研究员, 博士生导师. CCF 会员. 主要研究方向为网络流量管理与控制、数据中心网络、卫星网络.



Wang Shangguang, born in 1982. PhD, professor, PhD supervisor. Distinguished member of CCF. His research interests include service computing, mobile edge computing, cloud computing, and satellite computing.

王尚广, 1982 年生. 博士, 教授, 博士生导师. CCF 杰出会员. 主要研究方向为服务计算、移动边缘计算、云计算、卫星计算.



Ren Fengyuan, born in 1970. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include network traffic management and control, datacenter network, and IoT/industrial Internet.

任丰原, 1970 年生. 博士, 教授, 博士生导师. CCF 高级会员. 主要研究方向为网络流量管理与控制、数据中心网络、物联网/工业互联网.