基于直接拥塞通告的数据中心无损网络传输控制机制

胡晋彬[1,2],黄家玮[1],王建新[1],王 进[2]
(1. 中南大学计算机学院,湖南长沙 410083;2. 长沙理工大学计算机与通信工程学院,湖南长沙 410114)

摘 要: 数据中心网络广泛采用基于优先级的流量控制(Priority-based Flow Control, PFC)机制来避免因缓存溢出而丢包. 然而,PFC机制在保证无损传输的同时带来了队头阻塞和拥塞扩散等负面影响. 近年来,一些具备端到端拥塞感知能力的传输控制协议被提出来,有效缓解了网络拥塞,减少了PFC的触发. 但是在突发流量造成的瞬时拥塞场景下,这些研究工作仍会使得PFC频繁触发而导致严重的队头阻塞和拥塞扩散. 针对该问题,在端到端拥塞控制基础上,提出了一种实现于交换机上的直接拥塞通告解决方案(Direct COngestion Notification, DCON),该方案在突发拥塞场景下能及时识别出与非拥塞流(与造成拥塞无关的流)共享入端口的拥塞流(真正造成拥塞的流),并从交换机直接通告发送端对该拥塞流精确地降速. 实验结果表明,相比于现有的端到端拥塞控制传输协议,DCON有效避免了PFC的队头阻塞和拥塞扩散,平均流完成时间的最大降幅达到55%.

关键词: 数据中心无损网络;优先级流控;传输控制;队头阻塞

基金项目: 国家自然科学基金(No.62132022,No.62102046,No.62072056,No.61872387);湖南省重点研发计划项目(No.2022WK2005);湖南省自然科学基金(No.2022JJ30618,No.2021JJ30867,No.2020JJ2029);湖南省教育厅青年项目(22B0300)

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112(XXXX)XX-0001-12

电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20220491

A Transmission Control Mechanism for Lossless Datacenter Network Based on Direct Congestion Notification

HU Jin-bin[1,2], HUANG Jia-wei[1], WANG Jian-xin[1], WANG Jin[2]
(1. *School of Computer Science and Engineering, Central South University, Changsha, Hunan* 410083, *China*;
2. *School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, Hunan* 410114, *China*)

Abstract: Priority-based flow control (PFC) mechanism is widely deployed in data center network to avoid packet loss due to buffer overflow. Although PFC mechanism guarantees lossless transmission, it brings negative impacts such as head-of-line blocking and congestion spreading, etc. In recent years, many end-to-end congestion aware transport protocols have been proposed to effectively alleviate network congestion and reduce the triggering of PFC. However, in the case of transient congestion due to burst traffic, PFC is still triggered frequently even if the above end-to-end transport protocols are deployed, resulting in serious head-of-line blocking and congestion spreading. Therefore, on the basis of end-to-end congestion control, this paper proposes a direct congestion notification (DCON) solution implemented on the switches. DCON can timely identify the congested flows (really responsible for congestion) sharing the ingress port with the non-congested flows (not responsible for congestion). Meanwhile, DCON directly sends the congestion notification message to the corresponding senders from the switch and accurately sets the target rate for the identified congested flows at the sender. Compared to the existing end-to-end transmission control protocols, the experimental results show that DCON effectively avoids the head-of-line blocking and congestion spreading of PFC, and reduces the average flow completion time by up to 55%.

Key words: lossless datacenter network; priority-based flow control; transmission control; head-of-line blocking

Foundation Item(s): National Natural Science Foundation of China (No. 62132022,No. 62102046,No. 62072056,

收稿日期:2022-05-05;修回日期:2022-07-06;责任编辑:梅志强

No. 61872387); Key R&D Plan Projects in Hunan Province (No. 2022WK2005); Natural Science Foundation of Hunan Province (No. 2022JJ30618,No. 2021JJ30867,No. 2020JJ2029); Scientific Research Fund of Hunan Provincial Education Department (22B0300)

1 引言

近年来,为了降低数据中心内部网络传输延时,提高网络吞吐率,基于远程直接内存访问(Remote Direct Memory Access, RDMA)的RoCE (RDMA over Converged Ethernet)技术广泛部署于以太网数据中心网络[1~10](Data Center Network, DCN). 但是,在RDMA传输过程中,即使是单个数据包丢失也会大大降低网络吞吐率,使得流完成时间大幅增加,严重损害了应用服务的性能[11,12].

为了保证高效、可靠的RDMA数据传输,基于IP(Internet Protocol)和以太网的数据中心网络在链路层部署了基于优先级的流量控制(Priority-based Flow Control, PFC)机制防止缓存溢出,实现了无丢包的数据传输[13~15]. PFC机制是基于端口(或队列)的逐跳流控机制[16,17]. 当交换机入端口队列长度超过PFC暂停阈值,则向上游交换机发送PFC暂停报文,暂停上游交换机相关出端口的数据传输;当入端口队列长度减小到小于PFC的恢复阈值后,则向上游交换机发送PFC恢复报文,恢复其的数据传输,实现了无损传输,有效消除了丢包造成的重传延时.

然而,基于端口的PFC机制并没有区分出真正导致拥塞的流,PFC的暂停/恢复机制极容易导致队头阻塞、拥塞扩散和死锁等问题[18~20]. 当交换机入端口被其缓存队列第一个数据包的出端口暂停时,将导致队列中发送到其他出端口的数据包也被阻塞. 更严重的是,当网络中某个交换机发生拥塞,PFC逐跳流控机制最终会使得与该拥塞无关的上游交换机都会接收到拥塞信号并暂停数据包的转发. 拥塞不断向源端扩散会造成高排队延时和低网络吞吐率,大大增加了流传输时间.

因此,在部署了PFC机制的数据中心无损网络中,一些基于流的端到端传输协议相继被提出,以有效缓解网络拥塞,减少PFC的触发次数. 例如,基于显式拥塞通知(Explicit Congestion Notification, ECN)标记的DCQCN[1]协议和PCN[3]协议使用交换机上的ECN标记,通告发送端调整发送速率. 基于往返延时(Round Trip Time, RTT)的TIMELY[21]协议和Swift[22]协议在发送端根据测量的RTT调整发送速率.

以上基于ECN和RTT的传输协议采用了端到端的拥塞控制,其拥塞控制环路和更新周期都至少需要一个RTT. 例如,DCQCN的速率更新周期为50 μs[1]. 而且,这些端到端的传输协议通常需要多个RTT才收敛到目标速率,虽然能有效控制长流造成的持续拥塞,但难以控制生命期极短的突发短流所造成的突发拥塞. 因此,在突发拥塞场景下,即使部署了端到端的传输协议,也会不可避免地触发PFC,导致PFC队头阻塞和拥塞扩散等问题.

为了及时感知并控制数据中心无损网络的突发拥塞,以避免PFC队头阻塞问题,本文提出了一种基于直接拥塞通告的数据中心无损网络传输控制机制(Direct Congestion Notification, DCON). DCON保留了现有端到端拥塞控制方案的优点,能有效控制因长流竞争造成队列增长的持续拥塞;同时针对短流造成的突发拥塞,设计了快速拥塞感知方法,较好地解决了端到端传输协议难以控制突发拥塞的问题. 具体的,对于持续拥塞,DCON根据端到端的ECN标记调节发送速率;对于突发拥塞,DCON通过识别出与非拥塞流(与造成网络拥塞无关的流)共享入端口的拥塞流(真正造成拥塞的流),并从交换机直接通告拥塞流的发送端及时降低该拥塞流发送速率,有效避免了PFC的队头阻塞等负面影响.

本文主要的研究工作如下.

(1)通过真实网络测试床的实验,分析了现有应用于数据中心无损网络中的端到端传输协议在突发流量场景下难以避免PFC频繁触发,造成严重的队头阻塞和拥塞扩散的问题.

(2)提出了一种基于直接拥塞通告的传输控制机制DCON,在保留端到端传输机制有效控制持续拥塞的优点的同时,在交换机上识别与非拥塞流共享入端口的拥塞流,并将突发拥塞直接通告发送端,快速控制拥塞流.

(3)在可编程交换机和网络仿真平台NS-3(Network Simulator Version 3)上实现了DCON,并与现有数据中心无损网络传输协议DCQCN、Swift和PCN进行性能对比. 真实测试床和大规模仿真的实验结果表明,DCON有效避免了PFC的队头阻塞和拥塞扩散,将平均流完成时间和拖尾流完成时间分别降低了55%和64%.

2 相关工作

数据中心无损网络的传输控制机制是近年来学术界和工业界共同关注的研究重点,出现了一系列相关的工作成果. 依据不同的拥塞感知信号,数据中心无损网络的传输控制机制可总结为4类:基于量化拥塞通知的传输控制协议、基于显式拥塞通知的传输控制协议、

基于往返延时的传输控制协议和基于网络内部信息的传输控制协议.

(1)基于量化拥塞通知的传输控制协议,其代表性工作有QCN[23]和F-QCN[24]. IEEE 802.1 Qau工作组提出了量化拥塞通知(Quantized Congestion Notification,QCN)技术[18]. 交换机根据瞬时队列长度和期望平均队列长度的差异计算拥塞量化值,再根据拥塞程度概率性地将拥塞通知信号反馈到数据包的源端,以减小发送速率. 但作为链路层的流级别拥塞控制机制,QCN不能直接用于IP路由网络. 另外,由于QCN在网络发生拥塞时直接限制造成拥塞的发送端的发送速率,当发送端同时传输多条流时,QCN机制会损害到非拥塞流的传输性能. F-QCN[24]协议针对QCN的公平性问题,提出了一种公平量化拥塞通知方法,即交换机通过多播方式将拥塞信息反馈给所有拥塞源,公平调节发送速率. 但这种方法需要定制硬件来记录拥塞事件和计算采样频度,难以在数据中心广泛部署.

(2)基于显式拥塞通知的传输控制协议,其代表性工作有DCQCN[1]、DCQCN+[13]和PCN[3]. DCQCN协议是目前工业界在数据中心无损以太网络中采用最广泛的RDMA拥塞控制协议. DCQCN通过在IP包头中标记ECN,在传输层调节流的速率,避免了交换机上队列过长而触发PFC. 文献[2]对DCQCN在实际部署和运行过程中出现的拥塞扩散、死锁和活锁等问题进行了分析和改进,设计了一种基于区分服务的优先级流控机制来保证大规模部署. 针对DCQCN在大规模高并发场景下性能急剧下降的问题,DCQCN+协议提出了一种自适应调整探测带宽周期和步长的方法,使得速率控制能够适应网络的动态变化. PCN协议提出了一种新的拥塞探测和识别机制,通过识别造成拥塞的流和调节拥塞流的速率,有效地减少了PFC的触发. 然而,在流量突发强度增大时,由于上述基于显式拥塞通知的端到端拥塞控制至少要在第2个RTT才能对网络拥塞进行响应,难以快速控制网络突发拥塞而触发PFC.

(3)基于往返延时的传输控制协议,其代表性工作有TIMELY[21]和Swift[22]. TIMELY是第一个在数据中心内部采用往返延时RTT作为拥塞反馈信号的传输控制协议. TIMELY根据RTT的梯度变化动态调整发送速率,但由于其在动态网络下没有唯一的收敛点,有可能在很长的时间范围内收敛,使得其性能广受争议. 基于TIMELY在数据中心的部署经验,Swift使用网卡硬件时间戳精确测量RTT,并对不同路径和不同流的延时目标进行推理,达到了快速控制拥塞的效果,有效降低了拖尾流的完成时间. 然而,在基于RTT的传输控制协议中,延时反馈信号容易受到延时抖动或噪音的影响,导致拥塞检测的准确性降低.

(4)基于网络内部信息的传输控制协议,其代表性工作有HPCC[25]和P-PFC[26]. 在HPCC拥塞控制方案中,交换机使用了网络遥测(In-Network Telemetry,INT)技术,在每个经过的数据包包头添加当前时间、队列长度、历史发送数据量等信息;终端主机根据INT的反馈信息调整发送速率,进行拥塞控制. 相比于其他拥塞通知机制,HPCC能更精确的响应拥塞,但HPCC需要网卡和交换机的硬件技术支持,难以普遍适用于现有网络架构. P-PFC通过监控交换机缓冲区占用的变化率,预测PFC的触发时间并主动提前触发PFC的暂停机制,有效地控制了交换机缓冲区的占用比例,避免了数据流传输的拖尾现象.

总体上,上述针对数据中心无损网络设计的传输控制机制一部分是基于端到端的拥塞感知信号,难以及时应对突发拥塞;一部分是需要特殊技术支持,难以部署在基于IP和以太网的DCN中. 而本文提出的DCON既保留了端到端拥塞感知的优点,以有效缓解长流造成的持续拥塞,同时还设计了易于部署在交换机上的直接拥塞反馈机制,以及时处理突发短流造成的突发拥塞. DCON机制与现有方法相比,更快速的控制瞬时拥塞,减小了流的完成时间.

3 问题描述

本节首先介绍现有端到端的传输协议的工作机制,并通过试验床实验分析现有机制在突发流量场景下难以避免PFC队头阻塞问题的原因.

3.1 基于端到端拥塞感知传输机制的问题

目前工业界广泛采用的基于端到端拥塞感知的传输机制主要有DCQCN[1]和TIMELY[21],它们分别部署在微软和谷歌数据中心. DCQCN已被集成到迈络思ConnectX系列的网卡上. DCQCN的核心拥塞控制算法包括3个部分:(1)拥塞点(即交换机)根据瞬时队列长度进行ECN标记反应网络拥塞.(2)通告点(即数据接收端)将数据包的ECN标记复制到拥塞通告包,每50 μs生成一次通告.(3)反应点(即数据发送端)收到拥塞通告则根据ECN比例降低发送速率,在55 μs内未收到拥塞通告则增加发送速率. TIMELY则将RTT与T_{low}和T_{high}两个参数进行比较,将核心拥塞控制算法分为3个部分:(1)当RTT小于T_{low}时,增加发送速率;(2)当RTT介于T_{low}和T_{high}之间时,如果RTT梯度为正,说明网络拥塞加重,减小发送速率,反之则增加发送速率.(3))当RTT大于T_{high}时,减小发送速率.

下面通过试验床实验来验证现有利用ECN或RTT信号的端到端传输机制在突发流量场景下仍然频繁触发PFC,无法避免队头阻塞等问题. 考虑如图1所示的数据中心典型叶脊网络拓扑结构[4],其中包括3个叶交

换机(L0、L1、L2)、2个脊交换机(S0、S1)、16个发送端(H0~H15)和2个接收端(R0、R1). 在测试床实验环境中,每台主机(Dell PRECISION TOWER 5820)均配备10核 Intel Xeon W-2255 CPU、64 GB 内存和支持 DPDK(Data Plane Development Kit)Mellanox ConnectX-5 100 GbE NIC (Network Interface Controller),可编程P4(Programming Protocol-independent Packet Processors)交换机的缓存大小为22 MB,每个端口都启用了PFC机制且PFC PAUSE 阈值为320 KB. 所有链路带宽设置为40 Gbps,基础 RTT 为 40 μs. 传输控制协议部署的是DCQCN,参数设置为文献[1]中的默认参数,其中ECN标记阈值为200 KB.

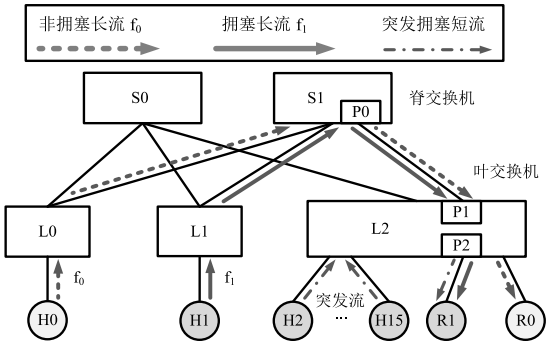
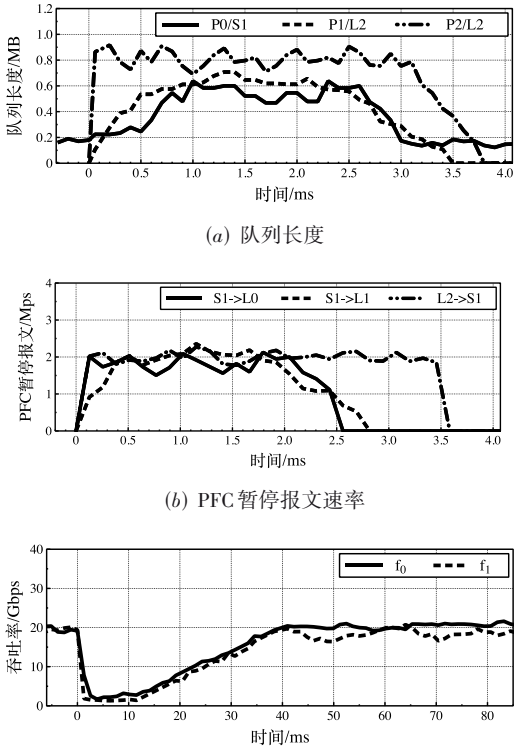


图1 数据中心叶脊网络拓扑结构



(*a*) 队列长度

(*b*) PFC暂停报文速率

(*c*) 实时吞吐率

图2 PFC与DCQCN相互作用的性能测试结果

两条长流 f_0 和 f_1 分别从发送端H0和H1发送到接收端R0和R1. 突发流量从发送端H2~H15发送到接收端R1,每个发送端生成35条大小为64 KB的短流,突发短流持续8 ms. 由于突发短流都以线速启动,每条短流在一个RTT内会发送完所有数据,端到端的传输机制无法控制这些突发流量. 由于 f_1 和突发流造成L2的P2端口出现拥塞,这些流都是拥塞流. 而 f_0 对于P2端口的拥塞无关,因此 f_0 是非拥塞流. 为了考察传输协议在突发流量场景下的性能,假设 f_0 和 f_1 在突发流量启动之前已经收敛到公平速率20 Gbps.

实验测试了DCQCN在上述常见数据中心网络突发流量场景下的性能,包括3个端口的队列长度、PFC暂停报文的传输速率(即单位时间内传输PFC暂停帧的数量)和两条长流 f_0 和 f_1 的实时吞吐率,测试结果如图2所示.

如图2(*a*)所示,由于 f_1 和突发短流到达同一个接收端R1,因此在突发流持续期间从L2到R1的P2出端口出现拥塞,P2/L2的队列瞬时增长很快. 由于端到端拥塞控制的延迟反应,f_1 无法及时降低发送速率,导致L2的入端口P1队列P1/L2也形成积压. 一旦P1的队列长度达到PFC阈值,L2则向其上游交换机S1的出端口P0发送PFC暂停报文,暂停P0端口的数据传输. 如图2(*a*)所示的P0/S1队列增长,导致拥塞扩散到了上游叶交换机L0和L1. 图2(*b*)显示了PFC暂停报文速率. L2到S1的PFC暂停报文导致PFC继续逐跳反压,即S1到L0以及S1到L1都持续有PFC暂停报文. 如图2(c)所示,在突发拥塞导致PFC暂停机制持续触发期间,f_0 和 f_1 的吞吐率下降到接近零. 当突发拥塞结束后,两条长流 f_0 和 f_1 逐渐收敛到公平速率20 Gbps.

3.2 基于端到端拥塞控制机制的问题分析

由上述实验分析可知,以DCQCN为代表的端到端传输协议难以及时控制突发拥塞,其原因主要有以下三个方面. 首先,发送端根据端到端反馈的ECN或RTT拥塞信号控制发送速率至少需要1个RTT的响应延时. 其次,发送端需要多个RTT才能将发送速率逐步调整收敛到目标速率. 最后,很多突发短流在不到一个RTT的时间内发送完所有数据包,无法通过端到端的传输机制控制它们的速率. 总之,在突发流量造成的瞬时拥塞场景下,PFC仍然频繁触发,带来了严重的队头阻塞和拥塞扩散.

因此,如果要避免因突发拥塞导致PFC队头阻塞,关键是要使拥塞流与非拥塞流共享的入端口不触发PFC,即需要及时识别出与非拥塞流共享入端口的拥塞流并及时降低该拥塞流发送速率. 为了实现这个目标,本文提出在保证该入端口不触发PFC之前,从交换机直接反馈拥塞信号到发送端并将拥塞流的速率直接降

到目标速率. 在如图1所示的突发拥塞场景下,如果能在L2的P1端口触发PFC之前,及时识别出经过P1端口的拥塞流f_1,并及时通告发送端H1将f_1的速率直接降为目标速率,就可以避免f_0在S0的P0端口经历队头阻塞.

4 DCON详细设计

本节介绍DCON传输机制的详细设计,描述DCON如何在交换机上直接感知拥塞并及时反馈到拥塞流发送端进行速率调节的传输机制.

4.1 设计思想

DCON是一种基于直接拥塞通告的数据中心无损网络传输控制机制. DCON针对突发拥塞设计了从交换机上直接反馈拥塞的通告机制并保留了端到端的持续拥塞通告,其总体结构如图3所示. DCON算法在终端主机和交换机上实现,由三个部分组成:发送端(反应点)、交换机(拥塞点/通告点)和接收端(通告点). DCON可以利用现有商用交换机的ECN和拥塞通知信号(Congestion Notification Message,CNM),不需要定制硬件支持,较容易部署.

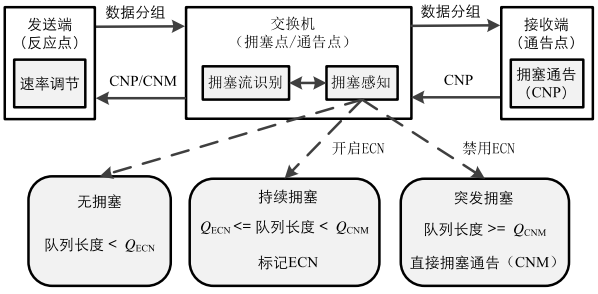


图3 DCON总体结构图

在交换机,DCON根据出端口队列长度检测端口的拥塞状态. 具体的,当出端口队列长度小于ECN标记阈值Q_{ECN},则认为该出端口无拥塞;当出端口队列长度介于ECN标记阈值Q_{ECN}和突发拥塞反馈阈值Q_{CNM}之间,则认为该出端口处于持续拥塞状态,对队列中超过Q_{ECN}的所有数据包进行ECN标记;当出端口队列长度超过突发拥塞反馈阈值Q_{CNM},则认为该出端口处于突发拥塞状态,此时禁用ECN标记,直到队列长度下降到Q_{ECN}以下再开启ECN标记,同时生成拥塞通告报文CNM,携带该出端口的拥塞流数量,直接发送到已识别的拥塞流发送端. 与现有端到端的传输控制机制相比,DCON的关键设计是可以从交换机直接通告突发拥塞到发送端,从而更快速控制拥塞流,避免PFC触发.

在接收端,DCON与DCQCN的机制类似,当收到有ECN标记的数据分组后,将ECN标记复制到拥塞通告分组(Congestion Notification Packet,CNP)中,周期性地向发送端发送CNP分组。在发送端,DCON收到CNP分组后,则采用加性增长/乘性降低(Additive Increase Multiplicative Decrease,AIMD)算法调节发送速率;DCON收到CNM分组后,将发送速率直接设为目标速率.

DCON适用于部署了PFC的数据中心无损网络中,通过交换机上的直接拥塞反馈机制可以快速调节拥塞流的发送速率,从而解决现有端到端传输协议难以控制突发拥塞而触发PFC导致队头阻塞和拥塞扩散问题. 由于DCON主要避免PFC的队头阻塞和拥塞扩散问题,DCON不适用于未部署PFC机制的通用网络和广域网络.

4.2 拥塞流识别

DCON的一个关键目标是避免突发拥塞场景下PFC的队头阻塞和拥塞扩散. 首先,DCON需要识别出经历突发拥塞的出端口,只要是到该拥塞出端口的流都被认为是拥塞流;然后,DCON需要识别出哪些拥塞流与非拥塞流共享了入端口. 如果某个入端口只有拥塞流的数据包,那么即使该入端口触发了PFC,也没有非拥塞流被阻塞,就不需要快速降低该拥塞流的发送速率.

如图4(a)所示,拥塞流f_1虽然是造成拥塞的流但不影响非拥塞流f_0,DCON不需要对f_1进行直接降速. DCON仅对于与非拥塞流共享同一个入端口的拥塞流进行直接速率控制,以避免非拥塞流因PFC暂停机制而遭受队头阻塞. 如图4(b)所示,DCON识别出拥塞流f_1与非拥塞流f_0共享入端口,因此将直接对f_1进行速率控制.

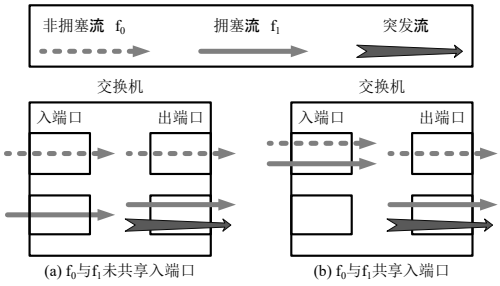


图4 DCON在交换机上识别拥塞流

具体的,在交换机上,DCON根据出端口队列长度识别突发拥塞. 当队列长度大于或等于突发拥塞反馈阈值Q_{CNM},触发突发拥塞直接反馈机制,DCON认为经过该出端口的流都为拥塞流,并在交换机上记录拥塞流数量N. DCON在交换机每个入端口记录了最后一个非拥塞流数据分组到达该入端口的时间T_{last}. 如果发生突发拥塞的当前时间$T_{current}$与T_{last}的间隔小于一个时间窗口T,则认为该入端口有非拥塞流经过,反之说明该入端口无非拥塞流经过. DCON仅对与非拥塞流共享

入端口的拥塞流进行直接速率控制，即仅对这些已识别的拥塞流发送端直接发送拥塞通告报文CNM.

4.3 拥塞通告与速率调节

DCON的拥塞通告包括两种情况. 第一种情况是当DCON识别出突发拥塞场景下的拥塞流后，则在交换机上生成拥塞通告报文CNM. CNM的报文格式如图5所示. 与非拥塞流共享入端口的拥塞流识别号和拥塞流数量N填充在CNM的QCN字段，然后将拥塞通告报文CNM直接反馈到拥塞流的发送端，以通告发送端调节拥塞流的速率. 该报文是利用现有商用交换机中部署的QCN机制的快速拥塞通告信号CNM[23]. 但由于QCN是基于L2地址转发数据分组，CNM报文中没有保留源端以太网包头，因此不能直接将CNM发送到基于IP路由网络的源主机. 为了解决这个问题，DCON在交换机的流表中记录源MAC地址和流识别号. 然后，交换机通过逐跳查找流表，更新目的转发地址，将拥塞流的CNM传输到相应的源端主机.

另一种情况是DCON在长流造成的持续拥塞场景下，

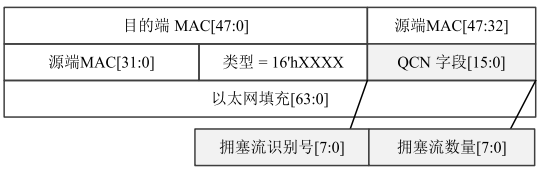


图5 拥塞通告报文CNM格式

即出端口队列长度大于或等于ECN标记阈值Q_{ECN}，且小于突发拥塞反馈阈值Q_{CNM}，则对队列中超过Q_{ECN}的数据包进行ECN标记. 当数据包到达接收端后，由接收端生成拥塞通告报文CNP并周期性地(默认为50 μs)发送到发送端.

发送端收到上述两种情况下的拥塞通告报文后，相应地调节发送速率. 发送端在以上两种情况下的速率调节如算法1所示. 发送端收到CNM后，DCON控制突发拥塞，将拥塞流的发送速率直接设置为目标速率C/N，其中C为交换机出端口链路带宽，N为拥塞流数量(第1行~第2行). 当发送端在50 μs内收到同一条流的多个CNM时，则只针对比当前速率更小的目标速率进行速率更新. 为了避免与持续拥塞的速率调节冲突，当交换机出端口队列长度超过Q_{CNM}后则停止ECN标记，直到队列长度减小到Q_{ECN}以下才重新启用ECN标记功能. 发送端收到CNP后，DCON根据算法1的增减速机制进行速率调节(第3行~第10行). DCON控制持续拥塞的速率调节算法中的参数是根据现有端到端传输协议DCQCN[1]的推荐参数进行设置的. 具体的，速率调节算法中的速率调节因子∂的初始值设为1，权重g设为1/256。DCON速率调节算法的时间复杂度为O(1)，不受网络规模的影响.

算法1 发送端速率调节算法

输入:突发拥塞通告CNM、持续拥塞通告CNP、当前发送速率、当前目标速率、拥塞流数量N、交换机出端口链路带宽C

输出:发送速率

```
1)  IF 接收到CNM
2)    发送速率=拥塞流的目标速率C/N;
3)  ELSE IF 接收到CNP
4)    IF 拥塞标记==1
5)      目标速率=当前发送速率;
6)      ∂=(1-g)∂+g;
7)      /*∂为速率调节因子,g为权重,设置为1/256 */
8)      发送速率=当前发送速率*(1-∂/2);
9)    ELSE
10)     ∂=(1-g)*∂;
11)     发送速率=(当前目标速率+当前发送速率)/2;
12)   END IF
13)  END IF
14)  RETURN 发送速率
```

值得说明的是，在数据中心典型的叶脊拓扑结构中，DCON从交换机直接反馈拥塞通告的方式使CNM最多经过3跳到达数据发送端，最快经过1跳到达发送端. 而端到端的拥塞通告CNP的传输，从拥塞点最多要经过7跳到达发送端，最快也需要经过5跳到达发送端. 因此，相对而言，DCON通过直接反馈能快速控制突发拥塞，同时又保留了端到端传输机制控制持续拥塞的优点.

4.4 突发拥塞反馈阈值计算

DCON在交换机上根据瞬时队列来判断网络拥塞状态. 当瞬时队列增长到突发拥塞反馈阈值Q_{CNM}则触发突发拥塞反馈机制，直接发送CNM到已识别的拥塞流发送端. 一方面，当突发拥塞反馈阈值Q_{CNM}较大时，有可能CNM还未到达发送端就触发了PFC PAUSE机制；另一方面，当突发拥塞反馈阈值Q_{CNM}较小时，过早发送CNM造成发送速率不必要的降低，导致瓶颈链路利用率不足. 因此，突发拥塞反馈阈值Q_{CNM}的选取变得很关键，既要保证CNM到达发送端时不触发PFC，又要保证瓶颈链路利用率高，同时不影响端到端的持续拥塞控制，即不小于ECN标记阈值Q_{ECN}.

假设在数据中心网络中典型的多对一叶脊网络拓扑中，多个发送端向同一个接收端发送数据，中间经过一个交换机. 链路带宽为C，网络中每一跳的延时是d. 每个发送端在时间t的发送速率为$v_i(t)$，交换机出端口在时间t的队列长度为$Q(t)$. 交换机入端口的PFC阈值为Q_{PFC}. 考虑在最坏情况下，交换机一个入端口的数据分组发送到M个出端口，则为了保证该入端口不触发PFC，某个出端口的队列长度不应超过Q_{PFC}/M. 当

DCON在T_C时刻触发突发拥塞反馈,此时的出端口队列长度为$Q(T_C)$, CNM最多经过3跳到达发送端,此时出端口的队列长度增长到$Q(T_C+3d)$,具体如式(1)所示:

$$Q(T_C+3d)=Q(T_C)+\sum_{i=1}^{M}\int_{T_C}^{T_C+3d}v_i(t)\,\mathrm{d}t-3dC \quad (1)$$

为了满足入端口不触发PFC的条件,即$Q(T_C+3d)<Q_{PFC}/M$,可得到突发拥塞反馈阈值Q_{CNM}的上限,如式(2)所示:

$$Q(T_C)<\frac{Q_{PFC}}{M}-\sum_{i=1}^{M}\int_{T_C}^{T_C+3d}v_i(t)\,\mathrm{d}t+3dC \quad (2)$$

DCON在实现中,突发拥塞反馈阈值Q_{CNM}保守的取值范围为$[Q_{ECN},\max(Q_{ECN},Q_{PFC}/M\text{-}3dC^*(M\text{-}1))]$.

5 DCON实现

本文在可编程P4硬件交换机和基于DPDK网卡的终端主机上实现了DCON算法.

DCON的拥塞流识别和拥塞感知功能实现在Wedge 100BF-32X可编程交换机上,用P4语言描述了数据包在数据平面的流水线处理过程. 一条流水线支持最多达12个数据处理单元. 数据处理单元将数据处理过程抽象成匹配和执行匹配-动作表的过程. 编译器将多个没有依赖关系的匹配-动作表排放在一个数据处理单元,并在不同数据处理单元之间排放有依赖关系的匹配-动作表. 一个数据处理单元内的匹配-动作表可以并行执行,但不同数据处理单元的匹配-动作表只能串行执行. DCON在交换机上的主要匹配-动作表包括从静态存储器(Static Random Access Memory,SRAM)读取队列长度,根据队列长度匹配无拥塞、持续拥塞或突发拥塞状态,再分别对应各状态的直接转发、标记ECN并转发和生成CNM并转发的不同动作. 拥塞流流表的每个条目包括16位流号和48位MAC地址,其中流号是通过循环冗余校验(Cyclic Redundancy Check, CRC)函数(CRC16)根据数据包包头的五元组(源IP地址、源端口、目的IP地址、目的端口和传输层协议)计算得到.

DCON的速率调节和ECN标记反馈是在终端主机基于DPDK技术实现的. 在发送端,应用层通过调用*send*()函数开始数据传输. 根据接收到的CNM或CNP相应地调节发送速率,再通过调用rte_eth_tx_buffer_flush()函数将数据包发送到网卡. 在接收端,通过调用rte_eth_rx_burst()函数从网卡缓冲区接收数据包,并调用receive()函数将数据包发送到应用层.

6 实验结果与分析

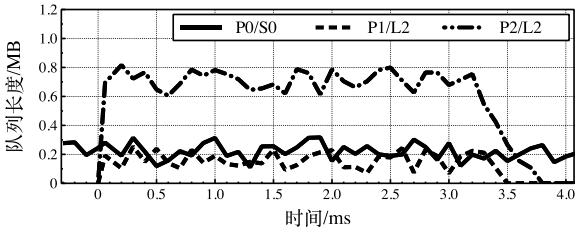
本文在真实网络测试床和NS-3网络仿真平台上分别实现了DCON. 为了对比性能,在测试床和NS-3中还实现了3个数据中心无损网络端到端传输协议DCQCN[1]、Swift[22]和PCN[3]. 首先,在基于3.1节中网络拓扑的测试床上,对DCON在突发流量场景下的基本性能进行测试;然后,在NS-3大规模网络拓扑结构中,测试了DCON在网页搜索和数据挖掘两种主流DCN应用场景下的性能. 最后,进一步测试了DCON在突发拥塞反馈阈值和拥塞流识别的时间窗口两个敏感参数下的性能.

6.1 测试床实验

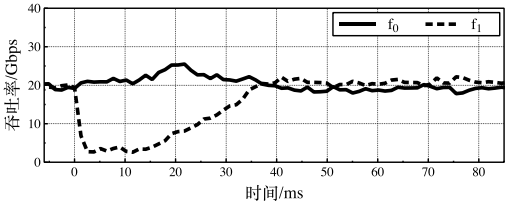
6.1.1 基础性能评估

首先对DCON在3.1节所述的突发流量场景进行基础性能评价. 实验参数设置与3.1节所述一致. 图6展示了3个端口的队列长度以及非拥塞流f_0和拥塞流f_1的实时吞吐率. DCON及时拥塞控制,有效避免了叶交换机L2的入端口P1触发PFC,非拥塞流f_0也没有经历队头阻塞.

从图6(*a*)和图6(*b*)可以看出,在突发流量存在期间,DCON拥塞感知和控制机制及时有效地控制了拥塞流f_1的速率. 非拥塞流f_0和拥塞流f_1共享的入端口P1/L2的队列长度未超过PFC阈值,避免了P0/S0端口被PFC暂停,非拥塞流f_0未受到阻塞. 总之, DCON有效地处理了突发拥塞,成功避免了PFC的队头阻塞和拥塞扩散问题.



(*a*) 队列长度



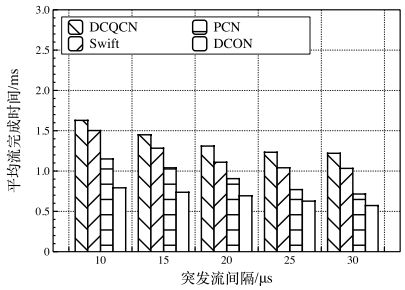
(*b*) 时吞吐率

图6 DCON基础性能测试结果

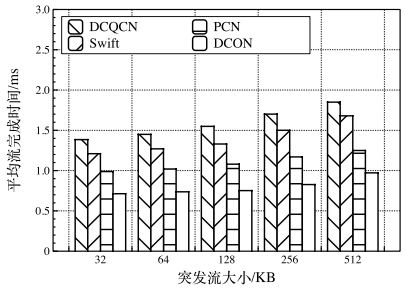
6.1.2 不同突发流量强度下的性能评估

本文进一步设计了改变突发流间隔和突发流大小的场景,以评估DCON在不同突发流量强度下控制突发拥塞的性能. 发送端H0~H1分别发送1条长流(200 MB)到接收端R0和R1. 发送端H2~H15同时发送短流到接收端R1,默认情况下每个主机发送30条短流,短

流间隔15 μs. 第一种情况改变突发流间隔,从10微秒增加到30 μs;第二种情况改变突发流大小,从32 KB增加到512KB. 两种情况下422条流的平均完成时间(即所有流完成时间之和除以流的数量)测试结果分别如图7(*a*)和图7(*b*)所示.



(*a*) 改变突发流间隔



(*b*) 改变突发流大小

图7 DCON在不同突发流量强度下的测试结果

从图7(*a*)和图7(*b*)可知,当突发流间隔减小或突发流大小增大时,突发拥塞增强突,发拥塞持续时间增大,所有机制的流平均完成时间都增大. 但是与其他三个端到端的拥塞控制机制相比, DCON能更好的控制突发拥塞,有效避免了PFC队头阻塞,从而获得了最小的流平均完成时间. 如图7(*a*)所示,在突发流间隔为20 μs时,DCON的流平均完成时间比DCQCN降低了47%.

6.2 NS-3仿真实验

6.2.1 大规模仿真实验

本文在数据中心叶脊网络拓扑下进行大规模NS-3仿真以评估DCON的扩展性能. 如图8所示,在该叶脊网络拓扑结构中,有8个核心层脊交换机和10个接入层叶交换机. 每个叶交换机与上层所有脊交换机相连,同时与24台终端服务器相连. 整个叶脊网络拓扑有240台服务器. 网络中所有链路带宽为40Gbps,每一跳的延时为5 μs.

网络流量在随机选择的主机对之间产生,且流量到达过程服从泊松分布. 本文选择数据中心的两种典型应用即网页搜索[26]和数据挖掘[27]来测试DCON性

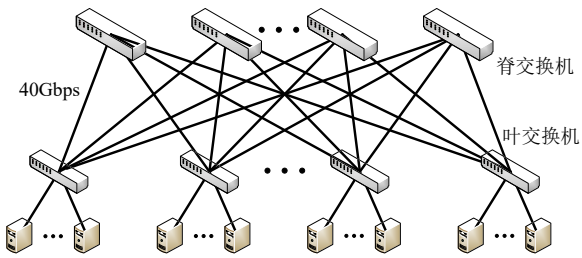


图8 大规模仿真的数据中心叶脊网络拓扑结构

能. 这两种工作负载都呈重尾分布[28],在网页搜索工作负载中,约60%的流小于100 KB,且约20%的流大于1 MB;在数据挖掘工作负载中约,80%的流小于100 KB,且约不到5%的流大于35 MB. 本文将网络核心层的平均负载从0.4逐渐增加到0.8.

本文以PFC暂停报文的速率和流完成时间作为主要的性能衡量指标[3]. 图9和图10分别展示了DCON在网页搜索和数据挖掘两种工作负载下的大规模仿真结果,包括PFC暂停报文速率、平均流完成时间和99分位拖尾流完成时间(即所有流完成时间分布中第99百分位的流完成时间).

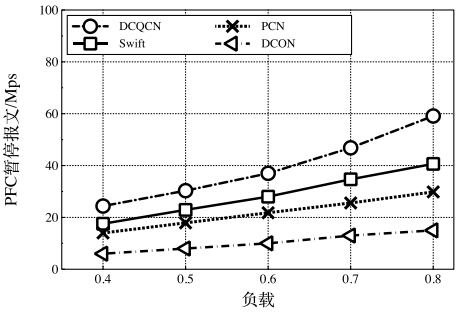
如图9(*a*)和图10(*a*)所示,与DCQCN、Swift和PCN相比, DCON有效减小了PFC暂停报文速率. 关键原因是DCON及时识别出拥塞流,并从交换机直接发送拥塞通告到相应的发送端,将发送速率直接降低到目标速率. 而DCQCN和Swift利用端到端的拥塞反馈不能及时应对突发拥塞,且发送速率要经过多个RTT才能降到目标速率. PCN虽然在1个RTT时间内将发送速率降到目标速率,但也是端到端的拥塞反馈机制,因此它们触发了更多PFC暂停报文的发送,尤其在有更多突发流量的数据挖掘应用场景下.

图9(*b*)和图10(*b*)展示了DCON获得了最低的平均流完成时间. 以网页搜索工作负载为例,相比于DCQCN、Swift和PCN, DCON在0.8的工作负载强度下,将平均流完成时间分别减少了55%、42%和20%. 图9(*c*)和图10(*c*)展示了DCON的99分位拖尾流完成时间最低. 实验结果说明DCON通过避免PFC的队头阻塞,有效减小了流的完成时间. 以数据挖掘工作负载为例,相比于DCQCN、Swift和PCN, DCON在0.8的工作负载强度下,将99分位拖尾流完成时间分别减少了64%、53%和32%.

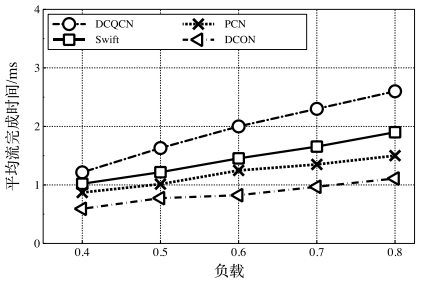
6.2.2 敏感参数测试

本文在6.2.1节大规模仿真的网页搜索工作负载下进一步测试了突发拥塞反馈阈值Q_{CNM}和拥塞流识别的时间窗口T对DCON性能的影响.

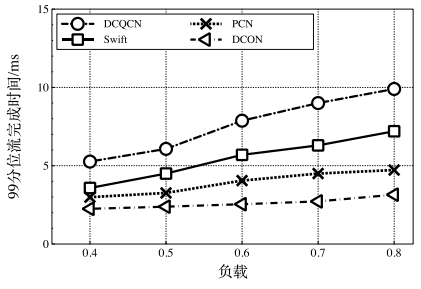
图11展示了在突发拥塞反馈阈值分别为40%固定缓存大小、60%固定缓存大小和根据4.4节计算的动态优化阈值三种情况下, DCON随着负载强度变化的平



(*a*) PFC暂停报文速率

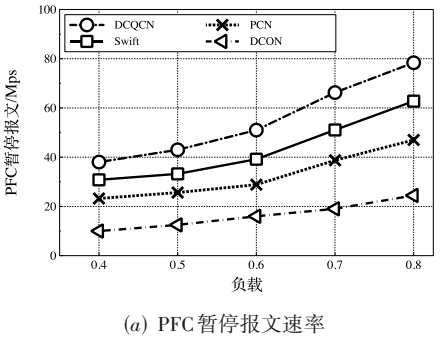


(*b*) 平均流完成时间

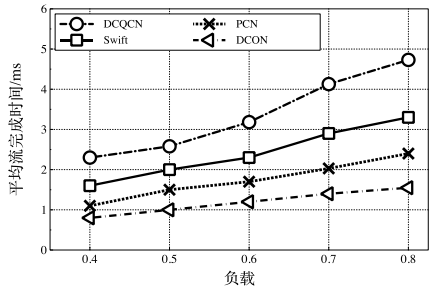


(*c*) 99分位拖尾流完成时间

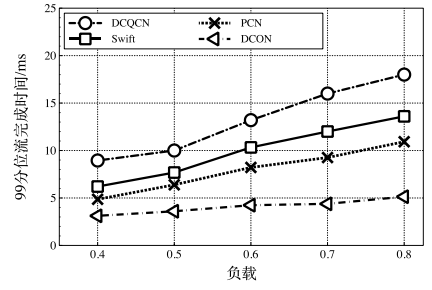
图9 DCON在网页搜索工作负载下的性能



(*a*) PFC暂停报文速率



(*b*) 平均流完成时间



(*c*) 99分位拖尾流完成时间

图10 DCON在数据挖掘工作负载下的性能

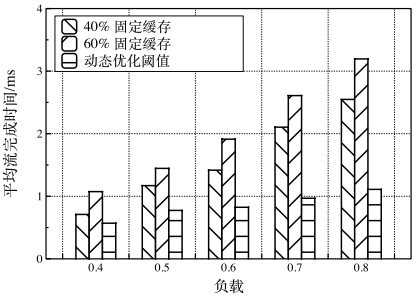


图11 突发拥塞反馈阈值的影响

均流完成时间. 由于突发拥塞反馈阈值较小时,过早的降低拥塞流发送速率,反而造成流完成时间增加;突发拥塞反馈阈值较大时,拥塞流的发送速率不能及时降低,导致PFC触发而使非拥塞流经历队头阻塞,流完成时间也会增加. 在动态优化的突发拥塞反馈阈值下,DCON的平均流完成时间最小,其性能最佳.

图12展示了识别拥塞流的时间窗口分别为2RTT、3RTT和4RTT三种情况下, DCON在不同负载强度下的平均流完成时间. 由于测试时间窗口小时,有可能非拥塞流还未结束传输而被误认为已经结束,与其共享入端口的拥塞流没有及时降速,导致PFC触发;当测试时间窗口大时,也有可能将已经结束的非拥塞流误认为还未结束,而激进地降低了拥塞流的速率,导致平均流完成时间增加. 从测试结果来看,拥塞流识别的时间窗口取合理大小的经验值3RTT时, DCON的平均流完成时间最小.

7 结束语

数据中心各类应用对低延时和高吞吐率需求的日益增加,也对数据中心网络的传输控制机制提出了更严苛的要求. 当前数据中心无损网络中采用的端到端传输控制协议无法控制突发短流,缺乏及时感知突发拥塞的能力,使得基于优先级的逐跳流控机制频繁触

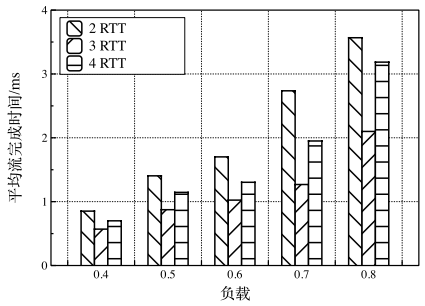


图12 时间窗口影响

发,导致PFC的队头阻塞和拥塞扩散等问题. 本文提出的DCON方案是一种在端到端拥塞感知的基础上,基于交换机感知并直接通告突发拥塞的传输控制机制. DCON既保留了传统端到端感知持续拥塞的优点,又增加了快速感知和反馈突发拥塞的能力,实现了良好的拥塞控制效果,且易于在交换机上部署. 基于测试床和NS-3仿真平台的实验结果表明, DCON有效避免了PFC队头阻塞,将平均流完成时间和99分位拖尾流完成时间分别降低了高达55%和64%. DCON利用交换机感知并直接反馈突发拥塞,解决了现有端到端拥塞控制的难点,为数据中心无损网络的传输控制机制提供了一种新的思路.

参考文献

[1] 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.

[2] 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, 2016: 202-215.

[3] CHENG W X, QIAN K, JIANG W C, et al. Re-architecting congestion management in lossless Ethernet[C]// Proceedings of the 17th Usenix Conference on Networked Systems Design and Implementation. Santa Clara: USENIX Association, 2020: 19-36.

[4] ZHANG Y R, LIU Y F, MENG Q K, et al. Congestion detection in lossless networks[C]//Proceedings of the 2021 ACM SIGCOMM 2021 Conference. New York: ACM, 2021: 370-383.

[5] 杜鑫乐, 徐恪, 李彤, 等. 数据中心网络的流量控制: 研究现状与趋势[J]. 计算机学报, 2021, 44(7): 1287-1309.
DU X L, XU K, LI T, et al. Traffic control for data center network: State of the art and future research[J]. Chinese Journal of Computers, 2021, 44(7): 1287-1309. (in Chinese)

[6] 李丹, 陈贵海, 任丰原, 等. 数据中心网络的研究进展与趋势[J]. 计算机学报, 2014, 37(2): 259-274.
LI D, CHEN G H, REN F Y, et al. Data center network research progress and trends[J]. Chinese Journal of Computers, 2014, 37(2): 259-274. (in Chinese)

[7] 王娟, 夏羽. TCP SkyLine: 数据中心网络高吞吐率传输[J]. 电子学报, 2020, 48(12): 2425-2433.
WANG J, XIA Y. TCP skyline: A high-throughput transport for data center networks[J]. Acta Electronica Sinica, 2020, 48(12): 2425-2433. (in Chinese)

[8] 崔子熙, 胡宇翔, 兰巨龙, 等. 基于流分类的数据中心网络负载均衡机制[J]. 电子学报, 2021, 49(3): 559-565.
CUI Z X, HU Y X, LAN J L, et al. Load balancing based on flow classification for datacenter network[J]. Acta Electronica Sinica, 2021, 49(3): 559-565. (in Chinese)

[9] 臧韦菲, 兰巨龙, 胡宇翔. 基于松弛时间与累计发送量的数据中心网络混合流调度机制[J]. 电子学报, 2019, 47(10): 2061-2068.
ZANG W F, LAN J L, HU Y X. Slack time and accumulation-based mix-flow scheduling in data center networks[J]. Acta Electronica Sinica, 2019, 47(10): 2061-2068. (in Chinese)

[10] 林智华, 高文, 吴春明, 等. 基于离散粒子群算法的数据中心网络流量调度研究[J]. 电子学报, 2016, 44(9): 2197-2202.
LIN Z H, GAO W, WU C M, et al. Data center network flow scheduling based on DPSO algorithm[J]. Acta Electronica Sinica, 2016, 44(9): 2197-2202. (in Chinese)

[11] LU Y W, CHEN G, LI B J, et al. Multi-path transport for rdma in datacenters[C]// Proceedings of the 15th USENIX Conference on Networked Systems Design and Implementation, Renton WA: USENIX Association, 2018: 357-371.

[12] 李文信, 齐恒, 徐仁海, 等. 数据中心网络流量调度的研究进展与趋势[J]. 计算机学报, 2020, 43(4): 600-617.
LI W X, QI H, XU R H, et al. Data center network flow scheduling progress and trends[J]. Chinese Journal of Computers, 2020, 43(4): 600-617. (in Chinese)

[13] GUO Z H, LIU S, ZHANG Z L. Traffic control for RDMA-enabled data center networks: A survey[J]. IEEE Systems Journal, 2020, 14(1): 677-688.

[14] 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.
ZENG G X, HU S H, ZHANG J X, et al. Transport protocols for data center networks: A survey[J]. Journal of Computer Research and Development, 2020, 57(1): 74-

84. (in Chinese)

[15] 邓罡, 龚正虎, 王宏. 现代数据中心网络特征研究[J]. 计算机研究与发展, 2014, 51(2): 395-407.
DENG G, GONG Z H, WANG H. Characteristics research on modern data center network[J]. Journal of Computer Research and Development, 2014, 51(2): 395-407. (in Chinese)

[16] GAO Y X, YANG Y C, CHEN T, et al. DCQCN: taming large-scale incast congestion in RDMA over Ethernet networks[C]//2018 IEEE 26th International Conference on Network Protocols (ICNP). Piscataway: IEEE, 2018: 110-120.

[17] TIAN C, LI B, QIN L L, et al. P-PFC: Reducing tail latency with predictive PFC in lossless data center networks [J]. IEEE Transactions on Parallel and Distributed Systems, 2020, 31(6): 1447-1459.

[18] HU S H, ZHU Y B, CHENG P, et al. Tagger: Practical PFC deadlock prevention in data center networks[C]// Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies. Incheon: ACM, 2017: 451-463.

[19] QIAN K, CHENG W X, ZHANG T, et al. Gentle flow control: Avoiding deadlock in lossless networks[C]//Proceedings of the ACM Special Interest Group on Data Communication. New York: ACM, 2019: 75-89.

[20] XUE J C, CHAUDHRY M U, VAMANAN B, et al. Dart: Divide and specialize for fast response to congestion in RDMA-based datacenter networks[J]. IEEE/ACM Transactions on Networking, 2020, 28(1): 322-335.

[21] 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.

[22] 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, 2020: 514-528.

[23] IEEE 802.1) IEEE. 802.1Qau Congestion Notification [S]. [2021-10-14]. http://www. ieee802. org/1/pages/802.1au.html.

[24] ZHANG Y, ANSARI N. Fair quantized congestion notification in data center networks[J]. IEEE Transactions on Communications, 2013, 61(11): 4690-4699.

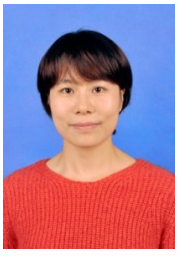
[25] LI Y L, MIAO R, LIU H H, et al. HPCC: High precision congestion control[C]//Proceedings of the ACM Special Interest Group on Data Communication. New York: ACM, 2019: 44-58.

[26] HU J B, HUANG J W, LV W J, et al. CAPS: Coding-based adaptive packet spraying to reduce flow completion time in data center[J]. IEEE/ACM Transactions on Networking, 2019, 27(6): 2338-2353.

[27] HU S H, BAI W, ZENG G X, et al. Aeolus: A building block for proactive transport in datacenters[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, 2020: 422-434.

[28] ROY A, ZENG H Y, BAGGA J, et al. Inside the social network´s (datacenter) network[C]// Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication. New York: ACM, 2015: 123-137.

作者简介



胡晋彬 女,1983年2月出生于湖南省长沙县. 现为长沙理工大学计算机与通信工程学院讲师. 主要研究方向为数据中心网络、RDMA网络和分布式机器学习系统. E-mail: jinbinhu@csust. edu. cn

黄家玮 (通讯作者)男,1976年10月出生于湖南省长沙市. 现为中南大学计算机学院教授、博士生导师. 主要研究方向为云计算、数据中心、软件定义网络、分布式机器学习、Web优化、流媒体、工业互联网. E-mail: jiaweihuang@csu. edu. cn

王建新 男,1969年12月出生于湖南省长沙市. 现为中南大学计算机学院教授、博士生导师、CCF高级会员. 主要研究方向为计算机算法与优化、网络优化理论、大数据应用、深度学习、生物信息学、虚拟实验环境. E-mail: jxwang@csu. edu. cn

王 进 男,1979年11月出生于江苏省扬州市. 现为长沙理工大学计算机与通信工程学院教授、博士生导师、CCF高级会员. 主要研究方向为移动自主网,车联网,无线传感网,物联网及应用. E-mail: jinwang@csust.edu.cn