DOI:10.3969/j.issn.1003-5060.2025.10.007

基于 RDMA 的高效拥塞控制方法设计

王芳慧[1], 黄正峰[1], 邱麟雅[1], 郭二辉[1,2]
(1.合肥工业大学 微电子学院,安徽 合肥 230601; 2.无锡众星微系统技术有限公司,江苏 无锡 214000)

摘 要:文章研究并解决数据中心的远程内存直接读取(remote direct memory access,RDMA)技术的网络拥塞控制问题。针对主流拥塞控制算法数据中心量化拥塞通知(data center quantized congestion notification,DCQCN)的收敛速度慢和缺乏硬件实现方案的不足,提出可参数硬件化的数据中心量化拥塞通知(parameterized DCQCN,DCQCN-p)算法,该算法通过优化拥塞流的速度因子 a、g 调整速度比例 R_c,并通过电路设计减少降速的频次;通过建立算法模型和搭建网络仿真 NS-3 平台,对比 DCQCN-p 算法在面临拥塞时单个调度流速度调整的性能以及多个调度流并发情况下的时延和吞吐量。仿真结果表明:在单个流面临拥塞时,DCQCN-p 算法的数据传输速率比 DCQCN 算法的提高了 50%;DCQCN-p 算法在链路上最小速率为 13.28 Gbit/s,相较于 DCQCN、TIMELY、数据中心传输控制协议(data center transmission control protocol,DCTCP)算法,分别增长了 24%、48%、23%;DCQCN-p 算法(方差 65%)的带宽分配公平性相较于 TIMELY 算法(方差 216%)和 DCTCP 算法(方差 191%)表现出显著的性能提升。

关键词:远程内存直接读取(RDMA);可参数硬件化的数据中心量化拥塞通知(DCQCN-p)算法;电路设计;多流高效;网络仿真
中图分类号:TN47 **文献标志码**:A **文章编号**:1003-5060(2025)10-1344-08

Design of efficient congestion control method based on RDMA

WANG Fanghui[1], HUANG Zhengfeng[1], QIU Linya[1], GUO Erhui[1,2]
(1. School of Microelectronics, Hefei University of Technology, Hefei 230601, China; 2. Wuxi Stars Micro System Technologies Co., Ltd., Wuxi 214000, China)

Abstract: This paper studies the network congestion control problem of remote direct memory access (RDMA) technology in data centers, and proposes a parameterized data center quantized congestion notification(DCQCN-p) algorithm to solve the problems of slow convergence speed and lack of hardware implementation schemes of the mainstream congestion control algorithm DCQCN. The DCQCN-p algorithm optimizes the velocity factors a and g of the congested flow to adjust the speed ratio R_c, and reduces the frequency of speed reduction through circuit design. By establishing the algorithm model and building the NS-3 simulation platform, the performance of the DCQCN-p algorithm in terms of the speed adjustment of a single scheduled flow and the delay and throughput of multiple scheduled flows in the face of congestion is compared. The simulation results show that the data transmission rate of the DCQCN-p algorithm is increased by 50% compared to the DCQCN algorithm when a single flow is congested. In addition, the DCQCN-p algorithm achieves a minimum link rate of 13.28 Gbit/s, representing a 24% increase over DCQCN, 48% over TIMELY, and 23% over data

收稿日期:2024-01-19;**修回日期**:2024-03-21
基金项目:国家自然科学基金资助项目(62274052;62374049);安徽省重点研究与开发计划资助项目(202304a05020003)和安徽高校协同创新资助项目(GXXT-2023-011)
作者简介:王芳慧(1996—),女,江苏南通人,合肥工业大学硕士生;
黄正峰(1978—),男,安徽无为人,博士,合肥工业大学教授,博士生导师;
郭二辉(1981—),男,安徽灵璧人,无锡众星微系统有限公司研究员,硕士生导师,通信作者,E-mail:guoeh@starsmicrosystem.com.

center transmission control protocol(DCTCP). The fairness of bandwidth allocation of DCQCN-p algorithm(65% variance) is significantly improved compared to TIMELY(216%) and DCTCP(191%).
Key words: remote direct memory access(RDMA); parameterized data center quantized congestion notification(DCQCN-p) algorithm; circuit design; multi-stream efficient; network emulation

0　引　　言

近年来，随着云计算和大数据的普及和爆发，数据中心已成为数字经济中不可或缺的基础设施。许多分布式应用部署在数据中心，如高吞吐量的分布式存储服务、延迟敏感型的网络搜索服务和数据库存储服务。对于需要低时延的应用程序而言，网络延迟至关重要。例如：网络延迟每增加 100 ms，亚马逊的购物网站将损失 1%的营收，谷歌搜索结果返回的数量会减少 0.2%～0.4%；而网络延迟每增加 400 ms，雅虎网站的流量就会减少[1]。

网络延迟大致分为终端主机上的网络协议栈延迟和网络内部延迟。为了降低协议栈的开销，远程内存直接读取(remote direct memory access，RDMA)技术将协议栈卸载到网络适配卡硬件中，避免了传统网络中数据从用户态到内核态的拷贝开销[2]，这一改进使得端到端的网络延迟从 50 μs 降低到 10 μs 甚至更低[3]。网络内部延迟主要是数据包的排队时延，当来自不同入端口的数据包需要经由交换机的相同出端口传输时，这些数据包会排队进入相同的队列[4-5]。若该出端口没有足够的可用容量来发送这些数据包，则会引发拥塞、数据丢失以及数据重传，从而降低链路上的吞吐量。

网络拥塞传播如图 1 所示。

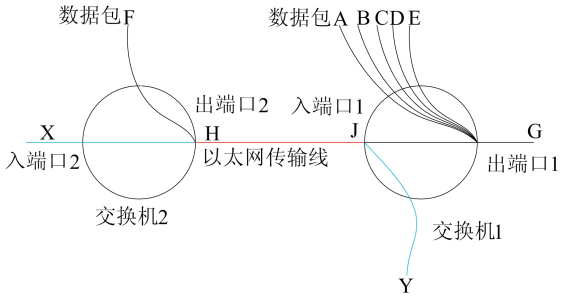


图 1　网络拥塞传播

国内外已有不少关于网络拥塞控制算法的研究。文献[6]以显式拥塞通知(explicit congestion notification，ECN)作为拥塞控制信号，并借鉴传输控制协议(transmission control protocol，TCP)的拥塞控制，优化了速率调整算法，提出了数据中心传输控制协议(data center transmission control protocol，DCTCP)算法；文献[7]提出了数据中心量化拥塞通知(data center quantized congestion notification，DCQCN)算法，将 QCN 算法与 DCTCP 算法结合并进行改进，通过参数配置实现更优越的控制，该算法已被应用在 Mellanox 公司的网卡产品中；文献[8]证明了数据包往返时间(round-trip time，RTT)可以作为网络中数据包拥塞的信号，并开发了 TIMELY 算法，利用现代网卡对时间戳和快速确认字符(acknowledge character，ACK)的支持，实施了基于精确 RTT 测量的拥塞控制；文献[9]在 P4 可编程交换机的基础上引入了拥塞通知的量化方法，提出了 P4 可编程的拥塞控制 (congestion control using P4-capable device，P4QCN) 算法，在轻微拥塞情况下实现带宽的平均利用率达到 90% 以上；文献[10]将 RTT 与 DCQCN 相结合，提出改进的 DCQCN (advanced DCQCN，DCQCN-A) 算法，该算法在 40 Gbit/s 链路带宽下实现了网络平均带宽利用率达 96%。

比较上文提到的几种网络拥塞控制算法[11]，得到各算法的优点和限制如下：

1) DCTCP 作为一种基于 TCP 的拥塞控制算法，虽然在某些性能方面表现出一定的潜力，但在充分利用交换机资源和提供队列优先级服务方面存在不足。

2) DCQCN 算法已成功在 Mellanox 产品中部署，并为网络提供了队列优先级服务，但它的速度调整相对较慢，且尚未提供相关硬件设计解决方案，这可能对网络性能构成挑战。

3) TIMELY 算法虽然具备潜在的性能优势，但其复杂的实现和对网络环境的敏感性(如网络干扰)可能导致性能下降。

4) P4QCN 算法尽管具备一定的潜力，但由于依赖可编程的 P4 架构，该算法尚未在大规模网络中成功部署。

5) DCQCN-A 算法是对 DCQCN 算法的进一步改进，但由于引入 RTT 和更多的算法系数，参数调整对性能易产生影响。

本文研究聚焦于融合以太网的 RDMA(RDMA over converged ethernet,RoCE)中的可参数硬件化的数据中心量化拥塞通知(parameterized DCQCN,DCQCN-p)算法、RoCEv2 技术,将以太网链路层取代 InfiniBand(IB)链路层,使其在更高层次的 3 层网络内进行通信[12-13];通过参数改进和硬件方案的设计优化算法,有效调整网络的发包速度,使降速和升速更为显著,从而进一步提高网络的吞吐量。相较于之前其他算法 40 Gbit/s 的网络部署,DCQCN-p 算法设定在 100 Gbit/s 和 25 Gbit/s 的混合网络中具有显著的带宽提升。

1 DCQCN 算法的基本原理

DCQCN 算法[7,14]需要网卡和交换机相互配合。交换机配置 ECN 来检测拥塞情况,网卡则进行源端和目的端的反馈和速度调整。算法分为下述 3 个核心内容。

1) 拥塞点(congestion point,CP)算法。交换机出口根据发送队列的长度来判断是否需要拥塞标记。

具体而言,有 2 个重要的队列阈值即 K_{max} 和 K_{min},用于衡量队列的大小。当队列长度 L_Q 大于 K_{max} 时,所有报文都会被标记为拥塞,概率为 1;当 L_Q 小于 K_{min} 时,所有报文都不会被标记为拥塞,概率为 0。在 L_Q 介于 K_{min} 与 K_{max} 之间的情况下,会根据一定的概率 P_{max} 来标记拥塞报文,计算公式为:

$$P = P_{max}\frac{L_Q - K_{min}}{K_{max} - K_{min}} \tag{1}$$

2) 通知点(notification point,NP)算法。网卡最多每 N 秒处理 1 个被标记为拥塞的数据包,并为该流生成 1 个拥塞通知包(congestion notification packet,CNP)报文。

3) 响应点(reaction point,RP)算法。目的端将拥塞信息反馈给源端,源端调整速度,具体如下。

当拥塞模块接收到 ECN 时,会执行降速操作,降速公式为:

$$R_t = R_c \tag{2}$$

$$R_c = R_c\left(1 - \frac{a}{2}\right) \tag{3}$$

$$a = a(1-g) + g \tag{4}$$

其中:a 为折减系数;g 为预配置常数;R_c 为当前速率;R_t 为目标速率。

当拥塞模块未接收到 ECN 时,根据设定的时间间隔内累加的超字节阈值次数 B、超时间阈值次数 T 与快速恢复次数 F 的比较结果,执行升速操作。

一般恢复时,B、T 都小于 F 时,升速公式为:

$$R_c = \frac{R_c + R_t}{2} \tag{5}$$

额外恢复时,B、T 任一个大于 F 时,升速公式为:

$$R_t = R_t + R_{ai} \tag{6}$$

$$R_c = \frac{R_c + R_t}{2} \tag{7}$$

其中,R_{ai} 为固定增加步长,可替换为 R_{hai},适用于 B、T 都大于 F 的超快恢复。

此外,需要在计时事件超时情况下才更新 a 值,即

$$a = a(1-g) \tag{8}$$

2 本文方法

2.1 改进的 DCQCN-p 算法

实际网络中可能会遇到数据校验错误、超时发送、路径错误等情况,解决方案有部署分布式机器学习的流量管理方案[15]、支持 TCP 协议的万兆以太网控制器[16]或在网卡上支持重传机制[17-18],支持回退 N 帧协议(go back N,GBN)。本文主要是针对 RoCE 中网络拥塞的情况进行分析,对于可靠报文进行重传机制,对于不可靠信息 UDP 协议只负责传输不支持应答。

网卡上的拥塞控制结合 DCQCN 算法进一步探索,主要在 NP 算法上调整 CNP 产生的时间(只要产生拥塞信号,就产生 CNP 报文)以及在 RP 算法中更新 a、g、R_c 计算,这些改进内容被整合到 DCQCN-p 算法中。DCQCN-p 算法的伪代码如下。

DCQCN-p 算法

1. if(首次收到 ECN 信息 ||(非首次收到 ECN 信息 &&. 连续同个流>降速间隔))
2. $T=0$, $B=0$
3. $R_t=R_c$(R_c 初值为 1,为首次全速比例)
4. $R_c=R_c(1-a/2)$ (a 初值为 1)
5. $a=(1-g)a+g$ (g 值为 0.062 5)(//RateDecrease())
6. else
7. T, B, R_t, R_c, a 保持不变 (//RateHold())
8. end if
9. if (时间阈值到期或连续同个流>=升速间隔)
10. T++

11. RateIncrease()
12. if (a<=0.5)
13. a=min{$a-1/25,(1-g)a$}
14. else
15. a=max{$a-1/40,(1-g)a$}
16. else if(降速间隔<连续同个流<升速间隔)
17. 最新的时间阈值需补偿(//减去这部分值：$T_{\text{最新完成阈值超时事件}}-T_{\text{最新收到CNP}}$)
18. RateDecrease() then RateIncrease()
19. end if
20. if(字节阈值到期)
21. B++
22. RateIncrease()
23. end if
24. if($T<F$ 或 $B<F$)(注：T，$B\neq 0$)
25. $R_c=(R_c+R_t)/2$ (//RateIncrease()的一般恢复)
26. end if
27. if ($T\geqslant F$ 或 $B\geqslant F$)
28. $R_t=R_t+R_{ai}$
29. $R_c=(R_c+R_t)/2$ (//RateIncrease()的额外恢复)
30. end if
31. 注：超快恢复 R_{hai} 在 if ($T\geqslant F$ 且 $B\geqslant F$)

在时间的改进方面，只要网络中标记了拥塞信息，就无间隔地产生 CNP 报文；ACK 报文信息只要在超时之前传输完毕，即可解析有效的拥塞信息。当网络需要精细地计算时间来控制速度时，会影响性能和增大硬件的设计面积。交换机设备需要计算网络中的排队时延和处理时延，网卡设备上需要记录发送点和接收点的时间反馈，这只对可靠传输报文记录 RTT 有效。

在算法中，需要设定一个拥塞间隔时间值，内部记录的是拥塞信息反馈到拥塞控制器的时间，对比下一次同个流的拥塞信息反馈到拥塞控制器的时间，若两者时间小于拥塞间隔，则说明拥塞标记频繁，实际还没进行调速反馈则舍弃；若两者时间大于拥塞间隔，则表明网络速度过快，需要进一步降速。对于时间阈值到期或连续 2 个流大于升速间隔，则需要累加超时间阈值次数 1 次；对于连续同个流大于降速间隔而小于升速间隔，则先默认降速再升速，在此基础上处理的时间需要补偿，即减去上一次流的完成时间与该流注册时间之差。

文献[19]提出了关于 DCQCN 算法在参数上的有关结论，即 DCQCN 参数的计时器行为是速率增加的主要因素，由于设置了大字节计数器，超快恢复阶段几乎不会发生。拥塞控制参数的对比见表 2 所列。

DCQCN-p 算法基于表 2 中参数进一步研究，使用拥塞控制算法时可忽略超快恢复阶段；此外，参考文献[7]、文献[20]和文献[21]中的 DCQCN 控制参数，改进的拥塞控制参数见表 3 所列。

表 2　拥塞控制参数对比

算法	DCQCN	QCN	小参数
字节阈值/MiB	10	0.15	0.15
时间阈值/μs	55	1 500	55
其他参数	K_{max}=200 KiB，K_{min}=5 KiB，P_{max}=1，g=1/256		

表 3　改进的拥塞控制参数

参数	DCQCN	DCQCN-p
字节阈值/MiB	10	10
时间阈值/μs	55	55
F	5	5
a	式(8)	步骤 10～步骤 15
g	1/256	1/16
R_c/(Gbit/s)	40	0～1
R_{ai}/(Gbit/s)	40	0.2
CNP 生成间隔/μs	50	按标记概率产生

建立降速、升速的速率模型[22]，简略地表达出关于 R_c 与 a 的关联，即

$$\frac{dR_c}{da}=\left(1-\frac{a}{2}\right)R_c'-\frac{a'}{2}R_c=-\frac{a'}{2}R_c \tag{9}$$

$$\frac{dR_c}{da}=\frac{(R_c'+R_t')}{2}a' \tag{10}$$

$$\frac{da}{dt}=1-g=0.937\,5 \tag{11}$$

$$\frac{da}{dt}=1 \tag{12}$$

g 的参考值有 2 个，即 1/256 和 1/16，1/16 相较于 1/256 速率抖动较大，因此这里使用 a 与 g 同时调节。

一方面，在降速过程中建议速率与 $a(t)$ 成负相关，在升速过程中建议速率与 $a(t)$ 成正相关。以 $\frac{a(t)}{2}$ 为分界，在 $a>0.5$ 时，使用升速较小的系数防止产生拥塞；在 $a\leqslant 0.5$ 时，使用升速较大的系数防止降到最低速率。

另一方面，为了节约计算资源，将接收到拥塞降速或达到算法要求升速的建议速率与 a 先进行计算，匹配出与 a 范围相关的值，并存入 RAM 中，速率则进行查表计算。

使用 DCQCN-p 算法在算法层面和拥塞参数上的改进方案，其拥塞控制模块的基本设计框架如图 2 所示。

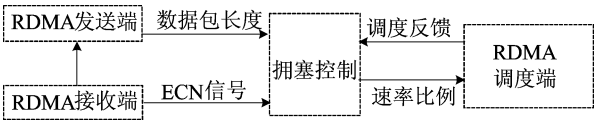


图 2 拥塞控制模块的基本设计框架

拥塞控制模块接收到来自 RDMA 接收端解析的拥塞标记后,会生成 CNP 包(对于不可靠连接和数据包类型)或带有后向拥塞标志的 ACK 包(对于可靠连接类型);随后,根据已累积的数据包长度和拥塞信息计算速率比例,并将相关速率比例发送给 RDMA 调度端,以控制调度端的发送字节数。

2.2 拥塞主控设计

拥塞主控单元由事件处理模块 EVENT_I_FIFO、地址映射表 CAM、参数存储表 PAR_TABLE 组成,并采用流水线设计且支持乱序存放网络拥塞流信息。拥塞主控设计如图 3 所示。

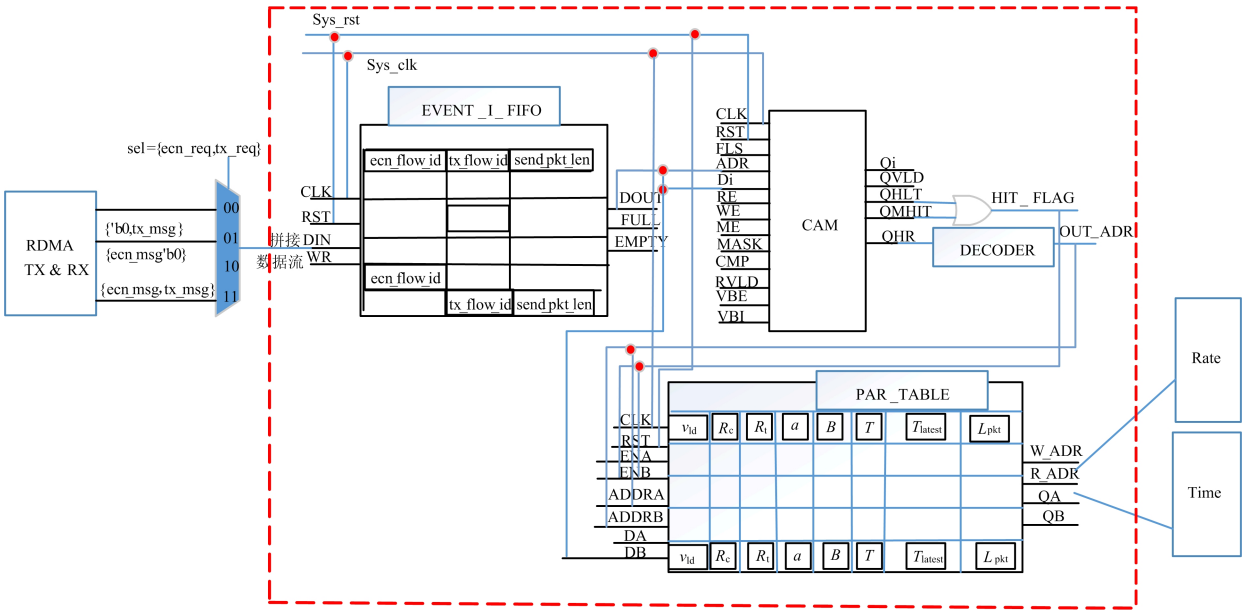


图 3 拥塞主控设计

EVENT_I_FIFO 用于接收 2 种不同来源的信息,即来自 RDMA TX 发送模块的发包信息(包括发包流编号和发包长度)以及来自 RDMA RX 接收模块的拥塞信息(拥塞流编号)。这些信息通过一个多路选择器与选择信号拼接成数据流,然后写入到队列中。只要队列非空,就可以 1 次读取 2 个数据项。

1) 检查这 2 个数据中的发包数据流编号是否一致,若一致,则将其数据长度累加聚合 L_{pkt}。

2) 比较这 2 个数据中的拥塞流编号是否相同,若相同,则将其中一笔拥塞信息丢弃。

3) 检查这 2 个数据中的发包数据流编号是否与拥塞流编号完全匹配,若匹配,则可以直接使用译码后的地址。

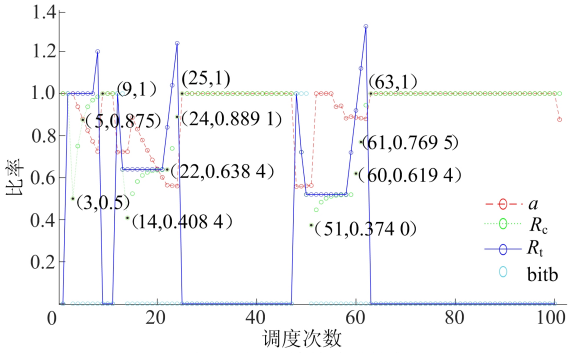
在地址映射中,以 4 000 个调度流为例,能够处理 512 个拥塞流,而超过 512 个拥塞流时则将数据包静默丢弃。地址映射的关键是将拥塞流编号放入内容寻址存储器(content addressable memory,CAM)中进行匹配。匹配过程利用了 QHIT+QMHIT 的逻辑“或”操作,若输出结果为 1 则表示控制器已注册过该拥塞流;若为 0 则表示没有注册,将拥塞流编号写到 CAM 中,并按照同样的过程来匹配下一笔拥塞流编号和下一笔发包数据流编号。然后进行发包数据流编号的匹配,若匹配失败则表示网络中此数据流不拥塞,将该数据包的长度静默丢弃;若匹配成功,则会保留该数据包的长度。

PAR_TABLE 用于存储拥塞控制模块的参数 v_{ld}、R_{c}、R_{t}、a、B、T、T_{latest}、L_{pkt}。其中:v_{ld}为拥塞流注册标志位;T_{latest}为拥塞流的最新注册时间。使用从 CAM 中译码的地址来更新拥塞流信息和字节信息,将这些更新的信息传输给 Rate 速率模块和 Time 超时模块。

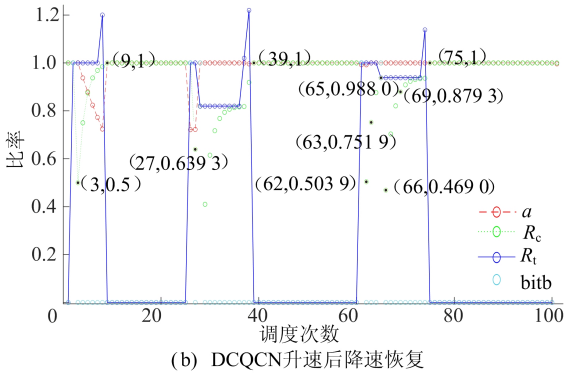
2.3 建模仿真

选择同一个调度流(queue pair number,QPN)的 100 次调度发包并设置标记概率 $P_{\max}=0.1$,进行 10 次拥塞标记(不丢包的拥塞处理),代表网络经历了 10 次拥塞事件。对于每一次的标

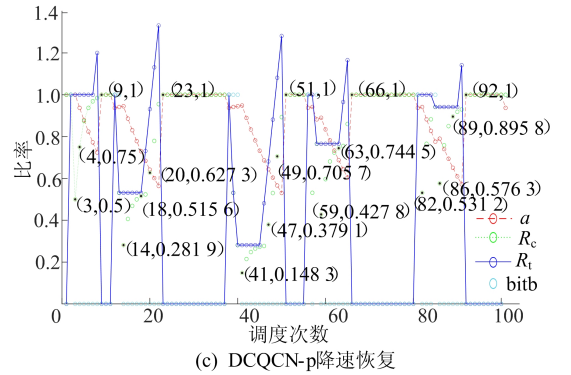
记，分别观察 1 次降速、2 次连续降速、3 次连续降速以及升速后再降速的情况，采用 MATLAB 仿真对比 DCQCN 算法与 DCQCN-p 算法的速度恢复情况，如图 4 所示。



(a) DCQCN 连续降速恢复



(b) DCQCN升速后降速恢复



(c) DCQCN-p降速恢复

图 4　DCQCN 与 DCQCN-p 的速度恢复对比

图 4 中：横轴表示 100 次调度发包；纵轴表示拥塞标志位 bitb 以及 R_c、a、R_t 在比率 0 ～ 1.4 内的变化，其中 bitb 只有 0 和 1 这 2 个数值，1 表示出现拥塞，0 表示未出现拥塞。

当首次出现拥塞信号时，图 4 中 R_c 为(3，0.5)，这意味着同个 QPN 的第 3 次调度发包的字节数为全速的 50%；当少于 5 次未出现拥塞信号时，系统将执行一般速度恢复策略，如图 4a 中的 R_c 为(5，0.875)和图 4c 中的 R_c 为(4，0.75)。然而，当连续 5 次未出现拥塞信号时，系统将执行额外的速度恢复策略，此时 R_c 为(9，1)，这意味着同个流在遭遇 1 次拥塞后第 9 次达到全速处理，此时 R_c 将恢复至 1，表明调度发包将以全速发送数据包，此时，R_t 的值将被设定为 0，标志着调度流速完全恢复后，将注销相关的调度流，直到下一次重新注册以进行速度调整。

由图 4 可知，DCQCN-p 算法可以在 100 次调度中处理包，而 DCQCN 则需使用 200 次调度才能实现同样的速度恢复；进一步比较可知 DCQCN-p 算法在连续 2 次或 3 次降速方面明显优于 DCQCN 算法，DCQCN-p 算法通常需要 9 到 10 次的恢复来达到全速状态，而 DCQCN 算法则需 11 到 13 次；此外，在降速后升速再降速的过程中，要将速度恢复到最大速度，DCQCN 算法需要调度 9 次，而 DCQCN-p 算法需要调度 6 次。总体而言，对于同个 QPN 发包，DCQCN-p 算法相较于 DCQCN 算法在速度恢复上快了 50%。

3　结果与分析

基于 NS-3 搭建点对点的网络拓扑结构，其中 80 台主机充当发送方，1 台主机担任接收端。仿真中，链路层传输路径设置为 100 Gbit/s 和 25 Gbit/s的混合网络；交换机参数配置为 40 个端口、4 个虚通道(virtual lane，VL)优先级和大小为 32 MiB 缓冲区。目标是验证 DCQCN-p 算法在局域网环境下如何应对多对一(many-to-one，incast)的突发性能挑战，并比较在多流并发情况下的时延以及吞吐量。为实现这一目标，借助 Linux 下的开源网络平台 NS-3[23] 框架构建网络平台，网络拓扑如图 5 所示。

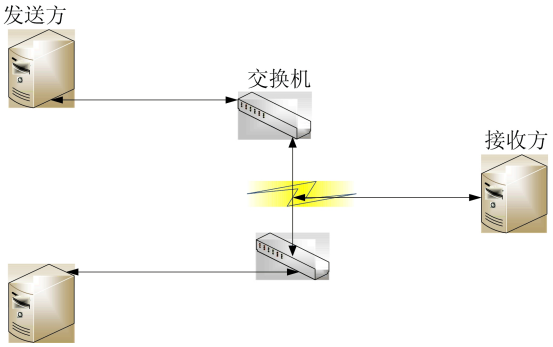


图 5　网络拓扑

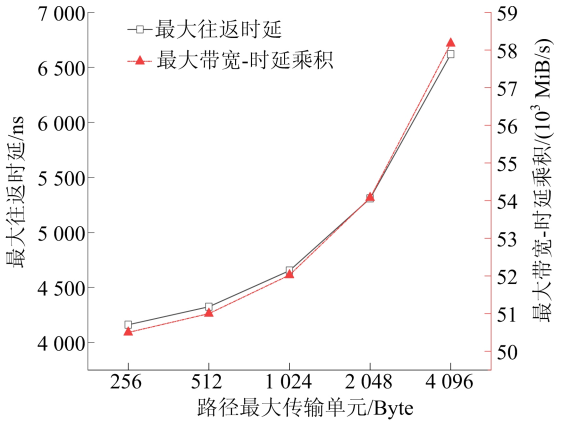
3.1　参数仿真分析

传输层携带的单次最大能够传输的有效数据负载即路径最大传输单元(path maximum transfer unit，PMTU)，影响网络的最大往返时延

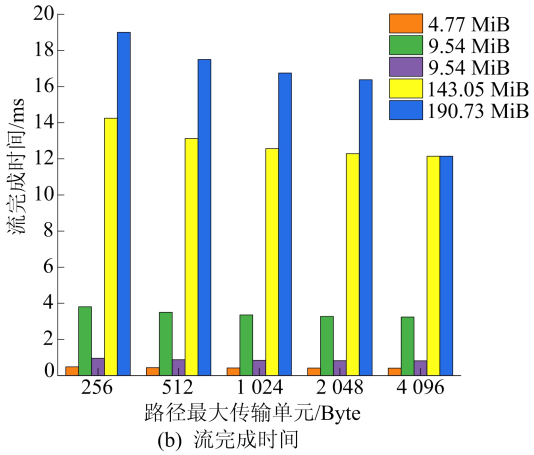
(round-trip time,RTT)和最大带宽-时延乘积(bandwidth-delay product,BDP)。PMTU 的取值为 256、512、1 024、2 048、4 096 Byte,网络性能有带宽(也称吞吐量)和时延(也称延迟)[24] 2 种基本度量方法。

网络流的完成时间(flow completion time,FCT)即网络带宽按照优先级严格分配将可以达到"接近最优的"网络流完成时间[25]。本文按照同一优先级发送数据包,总共需发送 972.89 MiB 数据量。其中:1 台主机发送 143.05 MiB 数据量;2 台主机分别发送 190.73 MiB 数据量;17 台主机分别发送 9.54 MiB 数据量;其余 60 台主机分别发送 4.77 MiB 数据量。在同一时刻传输数据,分别监测 80 个网络流各自的完成时间。

参数设置对性能的影响如图 6 所示。



(a) 最大RTT和最大BDP



(b) 流完成时间

图 6 参数设置对性能的影响

图 6a 中,在最大延迟和带宽传输效率之间找到平衡,使用 1 024 Byte 的 PMTU,NS-3 的仿真时间为 45.65 s,最大 RTT 为 4 654 ns,最大 BDP 为 52 025 MiB/s。一般情况下,发送数据的字节数与流完成时间成正比,图 6b 中出现了 2 类数据大小都为 9.54 MiB 的流,其流完成时间有差异是由于使用了不同的队列优先级,当数据流分配到专用的队列优先级时,流完成时间会降低。

综上,网卡设置字节阈值为 10 MiB、时间阈值 55 μs;交换机设置 40 个端口、4 个 VL,K_{min}=100 KiB,K_{max}=400 KiB,P_{max}=0.1,开启流控机制至少需要 4 000 个调度流,以满足网络最低速度需求。

3.2 控制算法的对比

使用 4 000 个调度流对 DCQCN-p、DCQCN、DCTCP 和 TIMELY 算法进行性能比较。DCQCN-p 算法还需设置其他的拥塞控制参数 R_{ai}=0.2、R_{hai}=0.8。各拥塞算法的性能比较见表 4 所列。

表 4 各拥塞算法的性能比较

算法	DCQCN	TIMELY	DCTCP	DCQCN-p
最小速率/(Gbit/s)	10.76	8.95	10.78	13.28
平均吞吐量/(Gbit/s)	18.74	23.48	25.08	23.69
incast 完成时间/μs	5 000	200	1 400	479

从表 4 可以看出:本文 DCQCN-p 算法在链路上最小速率为 13.28 Gbit/s,相较于 DCQCN、TIMELY、DCTCP 算法分别增长了 24%、48%、23%;DCQCN-p 算法的网络带宽分配的公平性方差为 65%,而 DCQCN、TIMELY、DCTCP 算法的分别为 15%、216%、191%。此外仿真中 DCQCN-p 算法比 DCQCN 算法多 3 次拥塞次数,这是由于主机从 100 Gbit/s 网络切换到 25 Gbit/s网络过程中速率降低引起的。

因此,相较于其他主流算法 DCTCP、TIMELY 和 DCQCN 算法,在局域网环境下,尤其是在 100 Gbit/s 和 25 Gbit/s 的多对一通信情境下,本文 DCQCN-p 算法表现出明显的突发性能优势。

4 结 论

本文提出一种在 DCQCN 的 NP 和 RP 方面的改进算法 DCQCN-p。仿真结果表明,相较于传统的 DCQCN 算法,DCQCN-p 算法在面临拥塞时成功提高数据传输速率达 50%;为了达到最大延迟和带宽传输效率之间的平衡,使用 1 024 Byte的 PMTU,最大 RTT 为 4 654 ns。相较于其他主流算法(DCQCN、DCTCP、TIMELY),DCQCN-p 算法表现出显著的性能提升,其最低速率高达 13.28 Gbit/s,incast 完成时延为 479 μs,网络带宽分配的公平性方差仅为 65%。

[参 考 文 献]

[1] BENSON T, AKELLA A, MALTZ D A. Network traffic characteristics of data centers in the wild[C]//Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. [S. l.]: IEEE, 2010: 267-280.

[2] 王凯. 低延迟数据中心网络中多应用通信优化机制研究与实现[D]. 南京：东南大学，2020.

[3] 石佳明. 数据中心网络中高吞吐低延时拥塞控制方法[D]. 北京：北京邮电大学，2022.

[4] NGDCN. Infiniband architecture volume 1, general specifications, release 1. 2. 1[M]. Beaverton: InfiniBand Trade Association, 2008.

[5] NGDCN. Infiniband architecture volume 1, general specifications, release 1. 3[M]. Beaverton: InfiniBand Trade Association, 2012.

[6] ALIZADEH M, GREENBERG A, MALTZ D A, et al. Data center TCP(DCTCP)[C]//Proceedings of the ACM SIGCOMM 2010 Conference. [S. l.]: IEEE, 2010: 63-74.

[7] ZHU Y, ERAN H, FIRESTONE D, et al. Congestion control for large-scale RDMA deployments[J]. ACM SIGCOMM Computer Communication Review, 2015, 45(4): 523-536.

[8] 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.

[9] GENG J, YAN J, ZHANG Y. P4QCN: congestion control using P4-capable device in data center networks[J]. Electronics, 2019, 8(3): 280.

[10] 胡永睿. 面向数据中心 RDMA 网络的拥塞控制算法研究及控制器设计[D]. 杭州：浙江大学，2022.

[11] 蒋万春，李昊阳，陈晗瑜，等. 网络拥塞控制方法综述[J]. 软件学报，2024, 35(8): 3952-3979.

[12] 刘军，韩骥，魏航，等. 数据中心 RoCE 和无损网络技术[J]. 中国电信业，2020(7): 76-80.

[13] 李家清，王祎玮，李道通，等. 智算中心 IB 及 RoCE 网络技术探究[J]. 电信工程技术与标准化，2024, 37(1): 42-48, 80.

[14] 张云泉，袁良，袁国兴，等. 2020 年中国高性能计算机发展现状分析与展望[J]. 数据与计算前沿，2020, 2(6): 1-10.

[15] YANG W, QIN Y, JIANG Z, et al. Traffic management for distributed machine learning in RDMA-enabled data center networks[C]//ICC 2021 IEEE International Conference on Communications. [S. l.]: IEEE, 2021: 1-6.

[16] 戴仕捷. 一种基于万兆以太网的 RDMA 的设计与实现[D]. 南京：东南大学，2020.

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

[18] ZHOU S, GONG Y, FAN Z, et al. SR-DCQCN: combining SACK and ECN for RDMA congestion control[C]//2022 IEEE 8th International Conference on Computer and Communications (ICCC). [S. l.]: IEEE, 2022: 788-794.

[19] SUGAHARA D, SHIRAKI O, YOSHIDA E, et al. A new DCQCN rate increase algorithm with adaptive byte counter [C]//2020 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR). [S. l.]: IEEE, 2020: 1-6.

[20] HU Y, SHI Z, NIE Y, et al. DCQCN advanced (DCQCN-a): combining ECN and RTT for RDMA congestion control[C]//2021 IEEE 5th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). [S. l.]: IEEE, 2021: 1192-1198.

[21] GAO Y, YAN Y, 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). [S. l.]: IEEE, 2018: 110-120.

[22] ZHAO X, LIU J, YAO J, et al. A time variant fluid model for DCQCN congestion control protocol[C]//2022 IEEE 22nd International Conference on Communication Technology (ICCT). [S. l.]: IEEE, 2022: 17-21.

[23] ZHU Y. NS3 simulator for RDMA over Converged Ethernet v2 (RoCEv2), including the implementation of DCQCN, TIMELY, PFC, ECN and shared buffer switch [N/OL]. (2016-10-25). https://github. com/bobzhuyb/ns3-rdma.

[24] PETERSON L L, DAVIE B S. 计算机网络：系统方法[M]. 5 版. 王勇，张龙飞，李明，等，译. 北京：机械工业出版社，2015: 25-26.

[25] 陆元伟. 数据中心内硬件资源高效的低延迟传输层研究[D]. 合肥：中国科学技术大学，2018.

（责任编辑　胡亚敏）