数据中心网络的流量控制：研究现状与趋势

杜鑫乐[1,2)] 徐恪[1,2,3)] 李彤[4)] 郑凯[4)] 付松涛[1,2)] 沈蒙[5)]

[1)](清华大学计算机科学与技术系 北京 100084)

[2)](北京信息科学与技术国家研究中心 北京 100084)

[3)](鹏城实验室 深圳 518000)

[4)](华为技术有限公司 2012 实验室 北京 100085)

[5)](北京理工大学计算机学院 北京 100081)

摘 要 作为海量数据快速存储和高效处理强有力的后盾，数据中心成为近年来学术界和工业界关注的热点。传统 TCP 难以在高吞吐、低时延、无损等方面同时满足当前数据中心传输需求，新的传输技术研究迫在眉睫。本文在对比传统 TCP 设计目标和数据中心网络中传输目标的基础上，对数据中心流量控制的研究现状展开综述。流量控制是指控制流量的发送速度以及发送规则，本文从基于端到端设计的拥塞控制和基于全局优化的流量工程两个方面对流量控制技术进行介绍，并从控制机制、扩展性、技术可行性等方面对上述技术进行了对比分析。最后本文对数据中心流量控制技术的未来研究趋势进行了总结和展望。

关键词 数据中心；流量控制；拥塞控制；流量工程；流量调度；负载均衡；

中图法分类号 TP393

Traffic Control for Data Center Network: State of the Art and Future Research

DU Xin-Le[1,2)] XU Ke[1,2,3)] LI Tong[4)] ZHENG Kai[4)] FU Song-Tao[1,2)] SHEN Meng[5)]

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

[2)](Beijing National Research Center for Information Science and Technology, Beijing 100084)

[3)](Peng Cheng Laboratory, Shenzhen 518000)

[4)](2012 Labs, Huawei Technology Co. Ltd., Beijing 100085)

[5)](School of Computer, Beijing Institute of Technology, Beijing 100081)

Abstract As a strong foundation for the rapid storage and efficient processing of massive data, the data center has become a hot spot in academia and industry in recent years. Traditional TCP is difficult to meet the demand for data center transmission in high throughput, low latency and loss-free aspects. Based on the comparison between the traditional TCP design target and the transmission target in the data center network, this paper summarizes the research status of data center traffic control. Traffic control refers to the control of traffic rates and sending rules. Therefore, this paper introduces the traffic control technology based on congestion control and traffic engineering and makes a comparative analysis of the above technology from the aspects of control

本课题得到华为技术有限公司委托项目(YBN2018065021)、国家自然科学基金(61825204， 61932016，61972039)、国家重点研发计划课题(2018YFB0803405)、北京高校卓越青年科学家计划项目(BJJWZYJH01201910003011)、北京市自然科学基金(4192050)、鹏城实验室大湾区未来网络试验与应用环境项目(LZC0019)资助.**杜鑫乐**，男，1996年生，博士研究生，主要研究领域为数据中心网络、网络协议设计.E-mail: dxl18@mails.tsinghua.edu.cn. **徐恪**（通讯作者），男，1974年生，博士，教授，博士生导师，主要研究领域为新一代互联网体系结构，网络空间安全与区块链系统.E-mail: xuke@mail.tsinghua.edu.cn. **李彤**，男，1989年生，博士，主要研究领域为网络协议、边缘计算和物联网. E-mail: li.tong@huawei.com. **郑凯**，男，1978年生，博士，主要研究领域为数据中心网络、SDN网络协议、广域网优化和物联网. E-mail: kai.zheng@huawei.com. **付松涛**，男，1982年生，博士研究生，主要研究领域为网络协议设计.E-mail: fust18@mails.tsinghua.edu.cn. **沈蒙**，男1988年生，博士，副教授，主要研究领域为云计算隐私保护、区块链技术与应用、网络传输协议. E-mail：shenmeng@bit.edu.cn

mechanism, expansibility and technical feasibility. Finally, this paper summarizes and looks forward to the future research trend of data center traffic control technology. According to the existing researches, we find that: (1) Considering the cost and performance, the most suitable traffic control algorithm for TCP/IP data center is DCTCP, and the most suitable traffic control algorithm for RDMA data center is DCQCN. Other researches require expensive custom hardware, which is difficult to deploy. (2) The traffic control technology is a technology of fair utilization of limited resources. Therefore, the performance of the technology can be improved by acquiring more relevant information or exchanging with other resources. E.g. ECN, RTT, traffic size, flow deadline. (3) Among the three main research points of congestion control, flow scheduling and load balancing, the mainstream algorithm system only focuses on one or two of them. (4) Smart NICs and programmable switches are widely used in the research of the data center network. The programmability of smart devices can bring new features to new technologies. In the end, the research directions are prospected. (1) A unified flow control test platform is needed. Different algorithms use different test environments, so it is difficult to evaluate them together. (2) Congestion control, flow scheduling, and load balancing studies need to be considered together. (3) Traditional distributed traffic control cannot be accurately scheduled due to insufficient information. As data centers grow in size, centralized controllers become network bottlenecks. The tradeoff between centralized and distributed control requires careful consideration. (4) High-performance programmable smart devices need to be developed and deployed. RDMA has become a hot topic in industry and academia. At the same time, programmable network devices greatly enhance the flexibility and rapid deployment of the network. (5) Traffic control design for specific application scenarios. The performance of the algorithm is improved by acquiring more relevant information or by exchanging other related resources. More resources are available in specific application scenarios. (6) With its strong self-adaptability and self-learning ability, artificial intelligence provides a set of effective decision-making tools for various research fields. The combination of artificial intelligence technology and network transmission technology is also a hot topic in the future. In summary, with the in-depth study of the data center, traffic control will become the most important basic performance tool for the data center, especially for the future high throughput, low latency requirements.

Key words data center; traffic control; congestion control; traffic engineering; flow schedule; load balance

1 引言

数据中心网络（Data Center Network，DCN）的出现为海量数据快速存储和高效处理提供了强有力的后盾。随着计算机互联网络的发展，数据中心已经成为国家和企业的核心基础设施。然而数据中心应用需求和模式的变化，给数据中心网络流量控制（Traffic Control）技术带来了巨大的挑战。这些变化有：

（1）随着数据中心内部的流量越来越高，用户使用网络的瓶颈从端云之间的网络传输逐渐移至云网络内部[1] [2] [3]。数据中心内部网络流量控制成为提升用户体验的关键因素之一。

（2）越来越大的网络规模，越来越高的流量负载[4] 使得数据中心网络区别于传统的网络。当前，超大规模的数据中心成为主导，业务模式从传统的租赁托管向提供云服务发生转变，成为新计算和新存储技术的主要消费者和采用者。大量的流量负载很难控制得当，即使调整少量流量对网络的影响也是十分巨大。因此流量控制在数据规模上也存在着重大的挑战。

（3）计算业务的时延敏感要求和存储业务的高吞吐要求冲突。对于计算业务来说，分布式计算提供了高效、可靠的性能，然而分布式计算的传输为网络带来大量内部短流量，短消息的传输成为计算业务的瓶颈；对于存储业务来说，基于数据中心的网络化存储提供了可扩展、高可靠的在线存储模式，然而数据存储业务超大的流量传输，常常影响其他业务的使用[5]。同时满足不同类别流量的低时延和高吞吐要求，是提高用户体验，提高云计算性能的必然要求。

传统的网络流量优化技术[6] [7] [8]，多是在路由层面进行流量均衡的优化，在数据中心这种特殊环境，如突发流量、低时延、网络流量的调度，传统

的流量控制技术已经很难满足云业务的需求[9] [10] 。简单的升级硬件并不能给网络的性能带来提升，反而有可能引起性能的衰减。因此，深入研究数据中心网络，研究在大规模高速的数据中心网络中，如何控制数据包在网络中的个数，如何控制流量大小和路由路径，如何在有限资源中提升网络利用率，对于提升网络服务能力和用户体验具有非常重要的价值和意义。

本文将详细介绍数据中心网络中有关流量控制的最新研究进展，对比分析现有流量控制的特点，总结并讨论数据中心网络流量控制的未来研究趋势。

2 问题与挑战

应用需求的发展使网络流量飞速增长，越来越多 SSD（Solid State Disk，固态硬盘）、10G 乃至 100G 交换结构以及光纤网络的使用已经成为数据中心网络发展的趋势[11] 。在这种情况下，数据中心流量控制技术将面临以下挑战：

（1）高吞吐

数据中心中，有关存储应用的数据流数目相对较少，但是它却贡献了最多的字节数[5] 。一般称这样的流为长流或者大象流（Elephant flow）。对于长流来说，它的时延需求不是很敏感，但是需要维持较高的吞吐率，以满足应用的需求[12] 。为了实现高吞吐率，需要有效、充分利用网络中的硬件转发能力，不能存在瓶颈环节。

（2）低时延

有关分布式计算或者 Web 服务请求之类的流量，它的数目在数据中心中占比非常高，但是每条流的长度却很短，只包含一些通知信息[13] 。一般称这样的流为短流或老鼠流（Mouse flow）。对于短流，通常希望快速返回结果，因此要求延迟最小化。这时数据排队往往成为瓶颈。不合理的传输策略往往是造成延迟增加的重要原因。

（3）Incast

Incast（又称 TCP Incast）是 many-to-one 的通信模式[14] 。这种情况发生在当一个父服务器向一组节点（服务器集群或存储集群）发起一个请求时，集群中的节点都会同时收到该请求，并且几乎同时做出响应，导致很多节点同时向一台机器（父服务器）发送 TCP 数据流，从而产生了一个“微突发流”。Incast 经常会导致交换机中的缓冲区溢出，从而发生流量崩溃[15] [16] 。

（4）优先级

传统网络中，由于每一个端节点都是一个用户，因此更加需要强调端节点之间的公平性；而作为一个整体的数据中心，则更看重整体对外的性能特性。在数据中心内部，往往需要优先满足更重要流的需求[17] 。如何针对不同的应用、不同的流量特性、不同的包类型设置不同的优先级，从而满足复杂的需求是数据中心面临的新挑战[18] [19] 。

（5）负载均衡

数据中心的主流拓扑结构一般采用带冗余的胖树拓扑[20] ，其目的是为了增加网络中可用等价链路，以提升网络性能，因而负载均衡成为数据中心的重要挑战。负载均衡是指合理的利用数据中心中的冗余链路，对不同的流量能够合理分配等价链路资源，从而充分使用网络资源，达到更好的用户体验。

3 研究现状分析

为了解决上述数据中心中存在的问题和挑战，研究人员在以下两个方面进行了探索和研究：拥塞控制（Congestion Control）和流量工程（Traffic Engineering）。拥塞控制的目标是缓解或者避免拥塞的发生，当拥塞发生时减小发送速率或者提前对于带宽进行分配，它可以一定程度上满足低时延要求，缓解 Incast 问题；流量工程是指根据各种业务流量的特性升级不同的传输策略，目标是同时满足优先级要求和负载均衡要求。

根据作用对象角度的不同，可以将流量工程进一步分为流调度（Flow Scheduling）和负载均衡（Load Balance）。流调度更加侧重对流的传输优先程度策略的设计，例如小流优先传输、截止时间流优先传输等；而负载均衡的目的是尽可能将流量在多条等价路径中分布，对网络资源进行充分的利用。数据中心流量控制技术现状，如图 1 所示。

为了提升传输性能，很多流量控制方案都会尽力在每个方向进行优化。本节将介绍数据中心网络流量控制技术背景以及现状。首先概述当前技术的设计理念，然后逐个介绍目前主流方案，最后进行总结。

3.1 数据中心传输技术背景

3.1.1 数据中心拓扑

一般的数据中心均采用胖树或者类胖树的 CLOS 拓扑结构[4] [5] 。图 2 展示了一个 k（4）叉胖树，自上而下分别为，核心层、汇聚层和边缘层。相比于传统的网络，数据中心明确的拓扑结构会提供给网络极大的传输潜力，而流量控制会使得这种潜力充分释放出来。

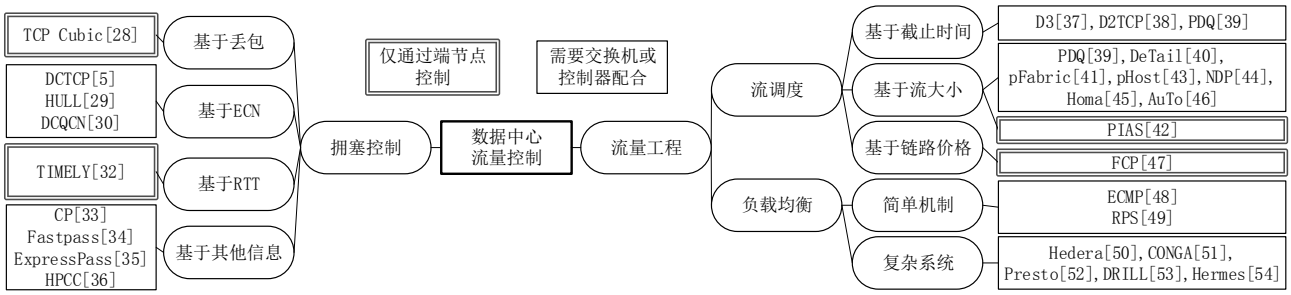


图 1：数据中心流量控制分类

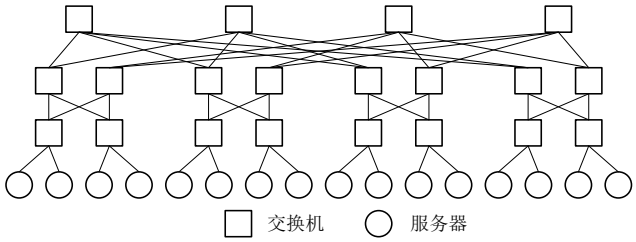


图 2：Fat-tree 拓扑结构

3.1.2 数据中心应用模式

常见的应用模式可以简单的分为两种，计算型和存储型。计算型一般使用如图 3 所示的 Partition-Aggregate 流量模式。

它一般面向用户的在线服务，例如，Google 或 Bing 的搜索结果，Facebook 的主页订阅等。Aggregator 接收用户的请求，并使用聚合树将其发送给 worker。在树的每个层次上，单个请求在不同的分区中产生活动。最终，worker 的响应被聚合并在严格的最后期限内返回给用户。它的流量模式常常为多对一。

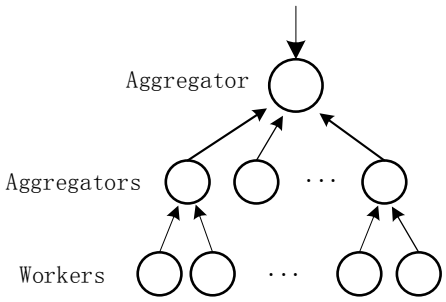


图 3：Partition-Aggregate 示意图

常见的存储型应用，如 Google 的文件系统（GFS）[21] 和 Hadoop 文件系统（HDFS）[22]，为了加快读取，增加冗余防止单点失效，常常采用多副本的策略。所以存储的流量模式也常常表现为多对一或者一对多。

多对一流量模式也称 Incast 流量模式，如何在大规模场景下缓解 Incast 问题，进行合适的流量控制，是目前数据中心中面临的重要问题之一。

3.1.3 数据中心流量控制性能指标

（1）吞吐量

吞吐量（Throughput）是指每秒接收到的不包括控制数据在内的比特数。有效吞吐量（Goodput）是单位时间内发到正确目标接口的比特数，如果减少了就丢弃或者重传。在计算机网络中，Goodput 是应用程序级的吞吐量，即来自某个源地址的网络转发到某个目的地的数目，不包括协议开销和数据包重传数据。例如，一个文件的有效吞吐量是指，文件的长度除以传输文件所需要的时间。有效吞吐量普遍低于吞吐量（总比特率传输物理），它通常低于网络接入连接速度（信道容量和带宽）。

（2）流完成时间

流完成时间（Flow Completion Time，FCT）是指一段时间，从发送第一个数据包开始(在 TCP 中，是 SYN 包)，直到收到最后一个数据包为止。当下载一个网页，传输文件，发送/阅读电子邮件，或者在几乎任何交互中涉及到网络时，用户希望他们的业务在最短的时间内完成，FCT 对于应用的感知程度是最大的，所以 FCT 对于衡量传输技术来说，是一个非常重要的指标。

（3）每包时延、往返时延

每包时延是指一个数据包从发送方发送开始，到接收方的收到为止，总共经历的时延，有时也称端到端时延。往返时延（Round-Trip Time，RTT）也是一个重要的性能指标，它表示从发送方发送数据开始，到发送方收到来自接收方的确认，总共经历的时延。它们都在包级别衡量传输技术的性能。同时它们也可以反映当前网络状态的拥塞程度。

（4）Slowdown

Slowdown 是完成一个 RPC（Remote Procedure Call）所需的实际时间与在无负载网络上完成 RPC 的最短时间的比值。Slowdown 值为 1 最为理想。这个指标一般用来衡量流优先级调度的效果，通过分析不同大小流提升可以感知流优先调度的效果。

（5）收敛时间

收敛时间是针对公平性指标的，它是指当多条流并发到达瓶颈段节点，多条流从不公平到公平的时间。理论上，收敛时间越短，公平性越高，性能越好。收敛时间可以通过测量，额外的流进入瓶颈

时，吞吐从不均衡到均衡的时间可以测得。

（6）交换机的丢包数

一般是衡量网络拥塞程度的指标。随着无损网络的引入，这个指标逐渐被弃用。一般拥塞控制能力越好，丢包数越少。

3.1.4 IEEE DCB 标准

为解决传统以太网在高拥塞期间丢弃数据包，造成重传使得许多应用无法忍受的问题，IEEE 制定了新的标准统称为 DCB（Data Center Bridging，数据中心桥接）。主要包括以下四个技术：

（1）基于优先级的流量控制

IEEE 802.1Qbb Priority-based Flow Control（PFC）[23] 是一种为数据中心提供无损网络的技术，所谓无损网络就是指网络不会因为拥塞而发生丢包。图 4 展示了一个在交换机层级之间实现 PFC 的一个实例。

PFC 实现无损的方式非常简洁，当交换机队列快满时（到达 ON/OFF 阈值），该交换机将会向上游交换机发送一个停止包，告知上游不要发送。等拥塞缓解，再通知上游继续发送。这样虽然解决了无损，但是如果有紧急的分组就无法发出，因此 PFC 通过虚拟队列，将数据包分成不同的优先级，即使某一优先级被拥塞阻塞了，仍可以通过更高优先级发送，以保证重要分组的及时传递。

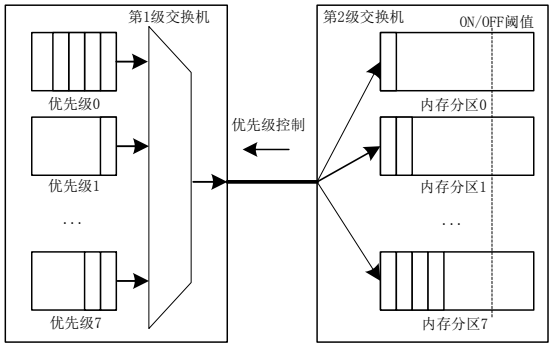


图 4：PFC 示例图

然而 PFC 如果要真实部署，仍然有许多问题需要解决，比如网络死锁，即当多个交换机同时成为另一个交换机的上级，他们可能同时通知对方停止发送，从而产生死锁，导致网络瘫痪。还有诸如公平性和拥塞扩散等一系列问题，在后文中介绍到的算法 DCQCN 一定程度解决了这些问题。在真实部署时还需其他机制进行配合。

（2）增强传输选择

IEEE 802.1Qaz Enhanced Transmission Selection（ETS）[24] 是为了保证网络业务带宽和\或限制带宽，针对性的提供差量服务。图 5 展示了 ETS 的一个实例。

这个例子使用 WRR（Weighted Round Robin）算法调度，它会按照权重轮询每个优先级队列，进行数据包的发送。还可以采用包括 SP（Strict priority algorithm）和 CBS（Credit-based shaper algorithm）等在内的其他算法。这些算法可以针对不同的优先级调整发送速率。

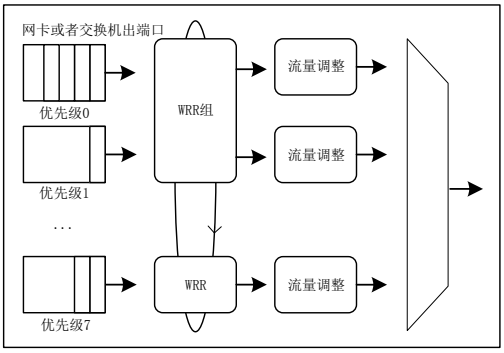


图 5：ETS 示例图

（3）量化拥塞通知

IEEE 802.1Qau Congestion Notification[25] 标准是为了解决热点拥塞的问题，也称为 Quantized Congestion Notification（QCN）。图 6 展示了一个 QCN 的实例。

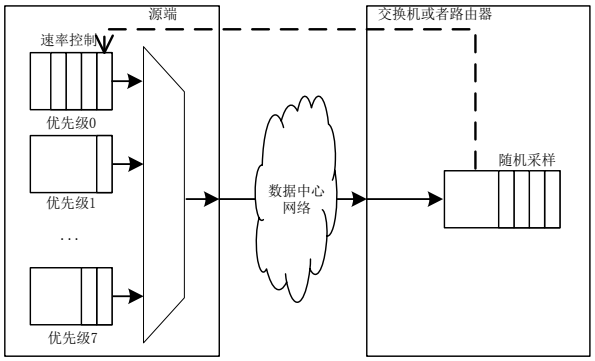


图 6：QCN 示例图

一旦中间交换机或者路由器发生拥塞，拥塞点会：

- 从交换机读出队列长度；
- 基于队列长度信息计算反馈的值；
- 格式化一个特殊反馈值的 QCN 帧，使用源 MAC 地址将该帧返回 QCN 源端；
- 根据 QCN 算法指定的动态信息更新队列的采样速率。

通过拥塞通告可以一定程度上解决拥塞热点的问题，然而在真实环境中很少实现它，因为它会高度依赖拥塞点反应时间，通过网络发送 QCN 帧的时间和反应点反应时间；并且它只能运行在二层网络上，很难适应数据中心大量的三层隧道功能。

（4）数据中心桥接交换协议

从上文可知，PFC、ETS 等都需要相邻设备之

间进行协调，为了在整个数据中心网络提供一致性的操作，DCBx（Data Center Bridging exchange）协议使用 IEEE 802.1ab[26] 交换标准实现相邻设备之间交换 DCB 功能的能力。

随着软件定义网络（Software Defined Network，SDN）的部署，DCBx 协议逐渐被淘汰，现在可以直接通过 SDN 控制器，对全网进行批量配置。

3.1.5 TCP 卸载和 RDMA 技术

目前 TCP/IP 在数据传输的过程中，需要消耗大量的 CPU 资源和总线资源，使得网络传输成为瓶颈，尤其是当 CPU 速度、内存速度和网口速度不匹配时，更加重了网络延迟效应。

TCP 卸载引擎（TCP offload engine，TOE）是一种用于网络接口卡（NIC）的技术，用于将整个或者部分 TCP / IP 堆栈的处理卸载到网络控制器。它主要用于高速网络接口，例如万兆以太网，其中网络堆栈的处理开销变得非常重要。

TOE 通常用于指代 NIC 本身。TOE 通常被建议作为减少与诸如 iSCSI 和网络文件系统（NFS）之类的互联网协议（IP）存储协议相关的开销的方法。

RDMA（Remote Direct Memory Access）是一种涉及数据中心里多种应用并且能改进数据中心性能的传输技术。这种技术是从 DMA（Direct Memory Access）发展而来的。RDMA 的设计思想是直接通过网卡访问对端内存，这样可以不消耗 CPU 资源。

RDMA over Converged Ethernet（RoCE）是一种网络协议，允许通过以太网进行远程直接内存访问（RDMA）。目前有两个 RoCE 版本，RoCE v1 和 RoCE v2[27] 。RoCE v1 是以太网链路层协议，因此允许同一以太网中任意两台主机之间的通信。RoCE v2 是一种互联网层协议，这意味着可以使用 IP 协议路由 RoCE v2 数据包。尽管 RoCE 协议受益于融合以太网网络的特性，但该协议也可用于传统或非融合以太网网络。

TCP 卸载和 RoCE 的相同点都是利用智能硬件，提升网络传输的性能，减小 CPU 的直接使用率。区别是 TCP 卸载是为了将 TCP 的一部分功能迁移到智能硬件上，如校验包头等；RoCE 的目的是为了远程复制与传输。RDMA 可以简单理解为利用相关的硬件和网络技术，服务器的网卡之间可以直接读内存，最终达到高带宽、低延迟和低资源利用率的效果。

新的硬件，新的技术可以更好地服务网络传输，然而如何更好地控制这些硬件和技术，充分发挥他们的作用，流量控制技术就显得尤为重要。

3.2 拥塞控制

所有网络设备的处理和转发能力都是有限的，一旦提供超过网络能够处理的极限，网络就会发生崩溃，进而导致大规模的拥塞和丢包。由于数据中心网络物理传输速率很高，拥塞导致的排队和丢包导致的超时重传会极大降低传输效率。因此，为了给数据中心提供更好的拥塞控制能力，研究人员在以下方面进行了深入研究。

3.2.1 基于丢包的拥塞控制

数据中心发展初期，其拥塞控制使用 TCP Cubic[28] 。Cubic 使用一个立方函数（cubic function）作为拥塞窗口增长函数。拥塞窗口增长与 RTT 无关，而仅仅取决上次发生拥塞时的最大窗口和距离上次发生拥塞的时间间隔值。

Cubic 算法的优点在于只要没有出现丢包，就不会主动降低发送速度，可以最大程度利用网络剩余带宽，提高吞吐量，在高带宽、低丢包率的网络中可以发挥较好性能。

Cubic 算法的不足之处是过于激进，没有出现丢包时会不停地增加拥塞窗口大小，向网络注入流量，最终将网络设备缓冲区填满，出现 Bufferbloat（缓冲区膨胀）。由于缓冲区长期趋于饱和状态，新进入网络的数据包会在缓冲区里排队，增加无谓的排队时延，缓冲区越大，时延就越高。另外 Cubic 算法在带宽利用率较高时依然在增加拥塞窗口，间接增加了丢包率，造成网络抖动加剧。

Cubic 是一个广域网拥塞控制算法，没有针对数据中心有任何特定设计，有经验的网络管理员会针对网络环境，对 Cubic 的参数进行调整。但对于算法本身并没有改进，仍使用丢包作为拥塞信号。

3.2.2 基于 ECN 的拥塞控制

Alizadeh M 等人于 2010 年提出 DCTCP (Datacenter TCP)[5] ，作为首次为数据中心设计的专用拥塞控制协议，DCTCP 认为降低传输时延的关键在于发送端能够根据网络实际状况，发出接收端能够正确接收的数据包。由于不需要大量数据缓存于交换机，传输错误时也不会引起超时重传，DCTCP 大大降低了传输时延。

DCTCP 设计了交换机和收发两端的调节机制，通过交换机队列长度识别拥塞程度并通过显式拥塞通知（Explicit Congestion Notification，ECN）接收端。接收端收到 ECN 标记后，回传带有 ECE（ECN-echo）标记的 ACK 回传发送端，发送端根据规则调节发送窗口，实现拥塞避免。相对 TCP 而言，DCTCP 所作的改动并不大，但十分切合数据中心网络的特点。DCTCP 不以丢包作为拥塞信号，而是以收到 ECN 标记为拥塞的信号。其核心是估计队列长度参数α，根据公式(1)计算：

$$\alpha = (1-g)\times\alpha + g\times F \quad (1)$$

式中 F 为当前窗口内包被标记 ECN 的比例，g（0□1）表示当前拥塞程度占总拥塞程度的权重。与传统 TCP 不同，出现拥塞时 DCTCP 并不将发送窗口减半，而是使用公式(2)调节，提升窗口恢复速度，提高吞吐量。

$$cwnd \leftarrow cwnd\times(1-\frac{\alpha}{2}) \quad (2)$$

DCTCP 基于数据中心网络流量调研，找准了引起数据中心网络传输时延增加的关键，摆脱了针对传统 TCP 协议参数调整这类修修补补工作的束缚，通过交换机与收发双方的结合，优化了数据中心网络拥塞控制性能，具有很强的实用性，因此被 windows server 2012 采纳。DCTCP 为研究适合数据中心网络传输的新协议提供了基础和方向，它利用数据中心网络节点要素间相互感知，使得共同优化传输协议成为现实。但 DCTCP 并没有区分数据中心网络的各类业务需求，从而不能提供具有优先级的服务；同时 DCTCP 对于交换机的利用并不是非常充分，在发送端数目特别多的情况下一样会导致交换机缓冲溢出，从而导致性能恶化。

Alizadeh M 等人于 2012 年提出 HULL (High-bandwidth Ultra-Low Latency)[29]，通过提前发现拥塞和为小流预留传输带宽，实现接近网络极限的低时延，它是 DCTCP 的改进。其控制位置主要位于交换机，收发两端也需要适当改动，其中网卡需要具备包速率调节功能。

HULL 的设计核心在于，如何为小流提供低的时延，减小排队时间。其关键在于，通过模拟一个低速率的交换机端口，来提前进行 ECN 标记。具体做法是，对每个交换机的出端口设置虚拟队列，虚拟队列只表示数据包的存在性，不保存包的具体信息，虚拟队列的出包速率小于真实队列的速率。进行 ECN 标记时，队列长度取决于虚拟队列的长度而不是真实队列。发送端采用 DCTCP 调整发送窗口，需要使用包速率调节(packet pacers)，其在发送端网卡实现，通过对数据包进行步频调节，以免大量突发占用交换机缓存，误传 ECN 信号，为了兼顾小流，对时延敏感的小流直接传送，不需要速率调节。

在小流大量增加的数据中心网络中，HULL 预留带宽过多必然导致网络性能下降，预留带宽过少又会增加小流时延，甚至引起超时重传，实用性降低。同时，HULL 需要对交换机和短接点网卡做出修改，真实场景中难以部署。

Yibo Zhu 等人于 2015 年提出 DCQCN[30]，它是一种基于速率的端到端拥塞控制协议，主要控制位置在网卡。DCQCN 认为基于远程过程调用（RDMA）的 RoCEv2[27] 传输技术栈可以代替传统的 TCP 协议栈，因为它能适应高带宽、低 CPU 开销和超低时延的环境[31]。

RoCEv2 采用 UDP 作为传输层的协议载体。由于 RDMA 的特性需要保证网络的无损性质，即网络传输过程中不允许丢包，因此需要基于优先级的流控制[23]（PFC）机制保证无损网络。但 PFC 直接应用于 RoCEv2 效果不理想，会有网络死锁和拥塞弥漫问题[26]，需要一种如 QCN[25] 一样针对流的拥塞控制策略，但 QCN 又无法直接应用于网络第三层。因此，DCQCN 对 QCN 进行了改造，结合 DCTCP 的控制策略实现拥塞控制。

DCQCN 算法包括发送端(reaction point, RP)、交换机(congestion point, CP)和接收端(notification point, NP)。交换机根据阈值记录网络拥塞情况，接收端网卡根据一定规则产生拥塞通知包(Congestion Notification Packets, CNP)，CNP 的产生按照时间间隔 N(比如 50 微秒)执行，一条流在之前的间隔 N 内未收到拥塞通知，当新的拥塞通知出现时，立即产生一条 CNP 分组并传送回发送方，之后每个间隔 N 内，接收方最多产生一条 CNP，以此降低网卡负担。发送端收到某条流的接收端回复的 CNP 后，按照类似于 DCTCP 的方式降低发送速率，首先将当前速率 R_C 保存为目标速率 R_T，再按公式(3)(4)降低速率：

$$R_C = R_C(1-\frac{\alpha}{2}) \quad (3)$$

$$\alpha = (1-g)\alpha + g \quad (4)$$

发送端如果在间隔 $K(K>N)$ 内均未收到 CNP 分组，则按照 $\alpha=(1-g)\alpha$ 更新 α 值；在发送速率的增加上，DCQCN 在 QCN 基础上做了改进，同时采用计数器和计时器的方式增加，计时器增加速率的方式能够实现快速恢复，比如在发送速率降到很低的情况下，经过了 5 次成功迭代的流，第一次可以用 $R_C=(R_C+R_T)/2$ 方式将速率提到目标速率一半以上，然后再按照加性增加规则增加速率。由于拥塞反馈和速率调整反应比较快，DCQCN 在此基础上进一步基于数据中心网络实际情况取消了慢启动过程，以提升网络效率。

DCQCN 对 QCN、DCTCP 做了符合数据中心网络的改造，即基于流实现拥塞控制，降低拥塞反馈时间，提高流启动速率和收敛速率，这种改造切合数据中心网络高带宽、高突发、低时延的需求，具有较强的实用性。但从 DCQCN 的实验数据来看，DCQCN 性能的优劣与参数选取强相关。例如，CNP 的产生间隔，计数器、定时器的设置，都需要在网络部署中具体确定最佳值，同时 DCQCN 与 TCP 并不兼容，也就意味着交换机需要区分两种流量，这些特点都决定了在大型数据中心网络中 DCQCN 的性能，还需要进一步验证确定。

3.2.3 基于 RTT 的拥塞控制

R. Mittal 等人于 2015 年提出 TIMELY[32] 拥塞控制协议，它通过发送端控制传输速率，是一种基于 RDMA 的端到端 RTT 测量和速率控制的数据中心网络拥塞控制算法。通过 RTT 测量、速率计算和速率控制三个模块，协同实现拥塞控制，如图 7 所示。

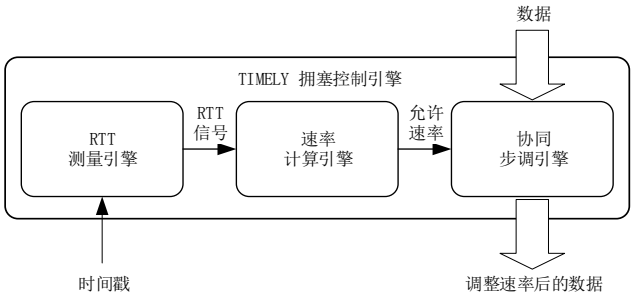


图 7：TIMELY 设计架构

RTT 测量是算法实现的基础，TIMELY 采用了反向信道 ACK 优先传送策略，保证 ACK 及时传送，同时由网卡直接产生 ACK，略去接收端产生 ACK 的时延，TIMELY 采用公式(5)计算 RTT，其中 $t_{completion}-t_{se}$ 为传输时间，交换机排队时延为 $\frac{seg.size}{NIC\_line\_rate}$。

$$RTT = t_{completion} - t_{send} - \frac{seg.size}{NIC\_line\_rate} \tag{5}$$

TIMELY 采用了三种拥塞控制策略调整速率，设置两个 RTT 阈值 T_{low} 和 T_{high}，低于 T_{low} 时，速率线性增加，高于 T_{high} 时，根据减速因子 β 降低速率；处于两个阈值之间时，则根据端到端时延变化斜率 $dRTT/dt$ 决策，当斜率小于等于 0 时线性增加，大于 0 时，根据 β 减小速率。RTT 阈值反应了数据中心网络的基本能力，时延变化斜率则记录了负载的动态变化，减速因子能够以切合数据中心网络需求的方式调整速率，计算得到的速率能够在数据中心网络中更好的传送。

有分组发送时，速率控制模块根据数据大小、规划速率等进行分段传输。由于有了速率信息，TIMELY 可以为每段数据定好时延，上一段数据传输完毕后，下一段数据直接传送，这就将以往基于窗口的发送模式改成了基于速率的发送模式，显然更适用于数据中心网络。

TIMELY 使用端到端的测量降低了对交换机设备的依赖，同时基于速率的控制比基于窗口的控制更适应低时延的数据中心网络。但 TIMELY 对网卡改动较大，且对硬件性能要求较高，实际中该算法对 RTT 的变化过于敏感，准确测量 RTT 十分困难。

3.2.4 基于其他信息的拥塞控制

Cheng P 等人于 2014 年提出 CP[33]，它是一个交换机上的辅助拥塞控制的机制。传统端节点使用 CUBIC 或者 DCTCP 等拥塞控制机制时，在面对超大规模突发流量时，队列发生丢包的现象是必然的。CP 认为传输时延大、吞吐损失等一系列问题的原因在于丢包之后，原来的 TCP 时钟状态丢失，只能通过超时进行重传，因此提出，在交换机即将拥塞时，截取数据包的包头，抛弃负载（Cut Payload，CP），从而继续维持原有的 TCP 时钟，使得 TCP 状态进行维持，迅速进行拥塞调整。而丢弃的负载也可以通过快速重传恢复，不必等待超时重传。

CP 从时钟状态出发，解决了由于超时重传造成的高时延、低吞吐，当遇到大量并发流时，CP 可以成功维持网络的高吞吐状态。然而 CP 需要对交换机的硬件增加功能，造成实际部署成本过大。

J. Perry 等人于 2014 年提出 Fastpass[34]，它试图将理论最优的算法用于实际系统。Fastpass 改变了以往通过收发端和交换机分布式解决时延问题的方式，采用集中控制的方式，从而真正实现全局最优。它在 DCN 中设置一个仲裁器(Centralized Arbiter)，所有发送端都需要与仲裁器交互信息，从而确定传送速率和路径，以此达到没有排队延迟，高带宽利用以及网络中流之间的资源共享。这种集中控制的方式类似于通过中心的导航系统为汽车导航，流能够选择最优的通行方式到达目的地。Fastpass 的结构如图 8 所示。

发送端有分组发送时，通过代理客户端向仲裁器发送自己需要传送的目的地和字节数，仲裁器规划好传输时隙和路径，传回发送端，发送端根据回传信息发送数据。显然发送端与仲裁器的信息交互、仲裁器的时隙和路径决策以及整个数据中心网络的时隙同步是 Fastpass 实现的关键。

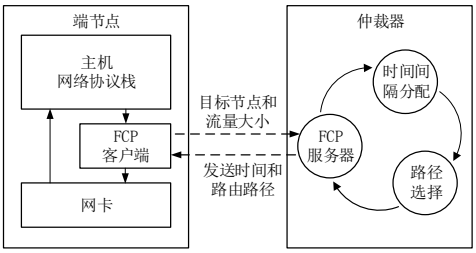


图 8：Fastpass 结构设计示意图

为使发送端与仲裁器信息交互更为流畅，设计了快速控制技术，具有低带宽消耗、低时延和高容错处理能力，保证发送端与仲裁器正常交互信息；仲裁器采用多核并行设备，通过流水线方式完成与节点通信、时隙分配和路径选择功能；通过 IEEE 1588 精确时间协议(PTP)同步节点时间，可避免由于操作系统调度导致的抖动。

节点与仲裁器之间信息交互的开销虽然在Fastpass的实验中仅需要1%左右，但实际部署中仍有较大开销； Fastpass认为单个仲裁器的处理能力在数百到数千个节点，当节点数量较多时，需要部署多个仲裁器协同处理，从而引起网络同步和低于1个时隙的数据流量需要聚合等问题，进而增加了实际部署Fastpass的难度。但是Fastpass可以作为一种理想的调度算法，理论上仲裁器能提供全局最优线路，使得它可以作为其它算法比较的基础，从某种角度说，一个算法只能无限接近理论上仲裁器规划带来的吞吐量和时延，而不能超过它。

I. Cho等人于2017年提出ExpressPass[35]，它是一个端到端的基于credit的拥塞控制协议。在发送数据包之前，ExpressPass使用credit包来预先探测拥塞，从而使数据传输能够保证有界延迟和快速收敛，并且可以应对burst的到来。与传统TCP不同的是，当需要发送时，首先需要向接收端请求credit，当接收端回传一个credit，发送端才会发送一个包。

ExpressPass利用交换机来限制credit的速率从而限制发送端速率。它的核心思想是将网络传输过程中正向拥塞通过交换机漏桶算法转换成反向credit的拥塞，同时通过对短小的credit进行拥塞控制，进而使正向网络不丢包，从而提升网络传输性能。它的本质是通过预先探测网络中的剩余带宽，进而可以准确确定发送速度。优点是提升短buffer可用性，降低丢包可能，减少重传，从而达到高性能。缺点是credit包的使用可能会过度浪费带宽。对于短流来说，本来直接发送即可，在ExpressPass中却需要等credit，并且有更大比例的credit被浪费，如何更精确地控制credit和分配是进一步研究的方向。

Yuliang Li等人于2019年提出HPCC[36]，它是一个基于RDMA的拥塞控制机制。HPCC利用in-network telemetry（INT）来获取精确的链路负载信息，并精确地控制流量。通过解决了在拥塞期间处理延迟的INT信息和对INT信息的过度反应等挑战，HPCC可以快速收敛、高效利用空闲带宽，同时避免拥塞，并可以保持接近零的网络内队列以获得超低延迟。

INT信息是当前传输端口的一些负载信息，包括时间戳（ts）、队列长度（qLen）、已传输的字节数（txBytes）和链路带宽（B）。通过这些信息，端节点可以准确计算当前网络中BDP（bandwidth-delay product）进而确定当前窗口需要变化的比例。从而提供一个准确的速率控制信息。HPCC的优点在于准确，缺点在于需要全网硬件均支持INT，包括交换机需要提供INT信息，网卡需要支持处理INT的能力，对增量部署不够友好。

3.2.5 拥塞控制小结

研究传输技术的重要目标之一是使流量达到满载。其中最重要的一点就是如何处理拥塞问题。因此如何获得更加快速，更加准确的拥塞信息成为一个新的趋势。

如表1所示，传统的TCP拥塞信号基于丢包；DCTCP通过1bit的标记表示交换机的拥塞状态；DCQCN通过PFC和ECN机制量化通知当前网络状况；TIMELY通过RTT的变换来确定拥塞；CP通过抛弃负载，通过包头维持拥塞状态；FastPass直接使用集中式的仲裁器，准确识别拥塞；ExpressPass用credit直接通知发送端带宽信息；HPCC使用了更加丰富的INT信息。

表1 更加准确的拥塞信息

名字	拥塞信息	优缺点
TCP Cubic[28]	基于丢包	丢包不可忍受
DCTCP[5] HULL[29] DCQCN[30]	ECN	广泛使用的拥塞信号
TIMELY[32]	RTT的变化	RTT的测量不准确
CP[33]	被切割的包头	交换机支持
FastPass[34]	集中式仲裁器	仲裁成瓶颈
ExpressPass[35]	使用credit直接通知发送端带宽信息	对原有协议栈改动大
HPCC[36]	INT	丰富的拥塞信息

未来拥塞控制主流的研究方向应当包括两个方面：如何更好更准确的利用当前已有的拥塞信号，如丢包、ECN和RTT；如何发现并使用新的拥塞信息。

3.3 流量工程

随着网络飞速发展，单纯提升网络传输速度不能满足人们日益增加的网络需求，如何在有限的资源下，给用户更好的体验，是当前研究的一个重要方向。流量工程是指根据各种数据业务流量的特性选取不同传输策略的处理过程。传统应用于广域网中的多协议标签交换（Multi-Protocol Label Switch，MPLS），由于技术相对复杂，构建成本相对高昂，因此在数据中心中未能广泛使用。而未来针对轻量化的流量工程或者端节点的轻量化部署成为研究热点。学术界针对数据中心中的流量工程，细化为两个方向，分别为流调度和负载均衡。

3.3.1 流调度

传统的网络流调度强调公平性，然而公平性并

不能有效提升用户体验，这一点在数据中心中尤为重要。因此针对不同的流量特性，提供不同的流优先级调度策略，是一个重要的研究方向。目前，有关优先级调度的研究分为以下几个方面。

3.3.1.1 基于截止时间的优先级调度

C. Wilson 等人于 2011 年提出 D3 (Deadline-Driven Delivery)[37]，它需要发送方、接收方和交换机共同参与优化，其核心思想是基于截止时间(deadline)实现各类流的差分服务。其设计包括：发送方依据剩余的数据量除以距离最后期限的剩余时间来计算所需带宽，并将请求的带宽信息填入包头。交换机接收这些数据包并提取带宽请求，根据已经分配的带宽状态，每个交换机在满足带宽请求前提下，尽可能将剩余带宽分配给有截止时间要求的数据流。对含有截止时间的数据流，其请求速率为 r，满足所有含有截止时间要求的流后，对所有流平均分配剩余带宽，得到速率 f_s，则最后数据流得到的分配速率为 $r+f_s$；对不含截止时间要求的流，则得到速率 f_s。交换机将允许分配的带宽信息写入包头并转发。分组转发路径上的每个交换机都执行以上相同的操作，并在包头创建一个向量，存放该交换机允许的带宽值。接收方将这个向量值拷贝到 ACK 包并发给发送方，发送方提取向量，选择最小的带宽值作为下一个 RTT 中将传送的数据数量。当没有足够可用的带宽时，D3 尽可能满足部分具有截止时间的流的需求，对不能满足需求的流提供一个基准速率(base rate)，允许发送端每隔 RTT 发送一个仅含包头的数据包。

D3 通过截止时间确定流的优先级，进一步量化了时延降低的标准，但当优先级相同的流用光网络资源时，D3 不具备优化性能的可能性；同时由于 D3 对于交换机的硬件改动较大，与 TCP 不能兼容，限制了其实用性。

Vamanan B 等人于 2012 年提出 D2TCP (Deadline-Aware Datacenter TCP)[38]，它结合了 DCTCP 和 D3 的优点，既考虑拥塞避免，也考虑流截止时间，同时兼顾到硬件和 TCP 的兼容性。与 DCTCP 一样，D2TCP 也需要交换机和收发两端共同配合，基于截止时间和拥塞程度完成发送端发送窗口的控制。D2TCP 主要做了两大改进，一是针对 D3 在交换机处集中分配的资源，发送端可以根据实际情况做出调整；二是对 DCTCP 的窗口调整策略做了微调，引入了截止时间因子 d，不再根据 α 调整窗口，而是根据 $P=\alpha^d$ 做出调整，调整策略也按照式（6）进行，在 P 等于 0 时加性增加窗口。

$$w=\begin{cases} w(1-\frac{p}{2}) & if\ p>0 \\ w+1 & if\ p=0 \end{cases} \quad 6)$$

显然，当 $d=1$ 时，算法和 DCTCP 一致，D2TCP 保持了与 TCP 的兼容性，发送方在计算 d 时充分考虑了流截止时间和网络拥塞程度，从而实现了相较 DCTCP 更好的性能。但作为一种分布式策略，D2TCP 的发送端在决策时只针对上一个 RTT 的拥塞程度和自身流需求，只能改善时延，无法对每条流进行流速控制，不能让数据中心网络网络针对全局做出最合理的流速规划，无法实现全局性的最早截止时间优先(Earliest Dealine First，EDF)和最短流优先(Shortest Job First，SJF)，限制了数据中心网络整体性能的提升。

C.-Y. Hong 等人于 2012 年提出算法 PDQ (Preemptive Distributed Quick)[39]，它通过分布于数据中心网络中的交换机实现全局范围内的 EDF 和 SJF，控制点同样包括收发端服务器和交换机，其控制核心在于交换机之间交互协作，从而达到近似集中控制的效果。PDQ 的设计理念如图 9 所示，在同时到达的流调度上，各类算法不能满足数据中心网络需求。图(a)所示为流的大小和截止时间，(b)为平均分配的情况，显然 f_A、f_B 均不能在截止时间完成，(d)为 D3 的基于截止时间的先来先服务策略，显然 f_A 也不能满足要求，而如果按照 EDF 或 SJF，如(c)所示，则所有流都可以在截止时间内完成。这说明在恰当的调度策略下，使用相同的资源能够实现更好的性能，这也是 PDQ 想要达到的目标，实现全局性的最优带宽配置。

PDQ 的发送端主要完成三件事情，发送数据时的变量确定、收到 ACK 后的变量更新，以及无法满足所有截止时间时的提早结束。发送数据时，对每个流确定一些变量，包括当前发送速率，流截止时间，对该流发出停止命令的交换机 ID 号等，当发送速率为 0 时，每隔一定间隔从交换机处探测速率信息；ACK 到达发送端后，发送端根据剩余的流大小、最新 RTT 值等更新变量；当网络容量无法满足所有流的截止时间时，发送端采用提早结束的方式，终结超过截止时限、无法在截止时限前完成或收到暂停信号且距离截止时限不足 RTT 的流。接收端将收到信号的头部复制到 ACK 中传送。在交换机处，采用 EDF 和 SJF 两种策略，对有截止时间的流采用 EDF，没有时限的流采取 SJF 策略，并辅以 ID 号、等待时间等策略，保证信号具备足够多的优先级。

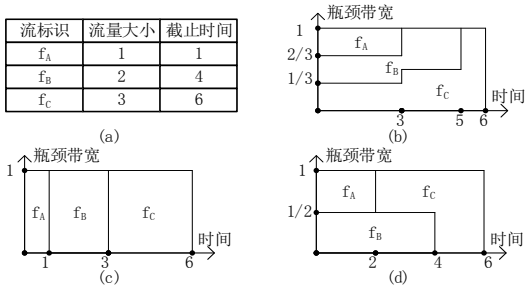


图 9：几种同时到达的流调度示意图

PDQ 通过交换机交互信息，维护每条链路的状态，保证数据中心网络流量传输在全局最优，力图在不改变网络结构，不增加中心处理环节的情况下实现类似于集中处理的性能，但兼顾每个流的需求特别复杂，对每条流的速率分配需要交换机掌握足够多的信息，同时需要协调所有交换机参与，这对动态变化的数据中心网络而言难于实现，导致难于在现实中部署。

3.3.1.2 基于流量大小的优先级流调度

D. Zats 等人于 2012 年提出 DeTail (Reducing the Flow Completion Time Tail)[40] ，它认为如果一条流因为丢失重传等原因不能尽快传完，就会形成比较长的尾部(Tail)，增加 DCN 传输时延，因此需要采取某种策略让每条流尽快传完。与通过改进传输层控制算法不同，DeTail 是一个跨层配合的方案，利用 DCN 高速和多径的特点，通过链路层、网络层、传输层和应用层共同完成两项工作：对较长的流，减少包丢弃；对时延敏感的流进行优先级排序，平衡网络负载。

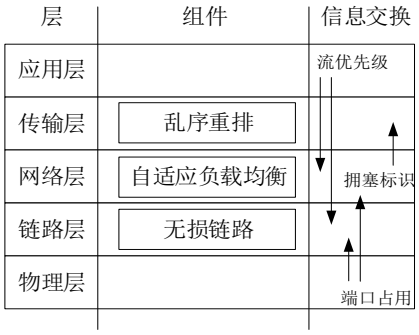


图 10：DeTail 的跨层协议配合图

DeTail 对协议栈的改进如图 10 所示，链路层采用缓冲建立一个无损环境，网络层基于负载均衡为每个包选择路径，避免拥塞；基于链路层和网络层协议，包不会因为拥塞丢失，传输层将端口占用情况作为拥塞通知，在应用层设置流的优先权，区分时延敏感与不敏感的流，保证敏感的流不被低优先级的流阻塞。显然 DeTail 的控制位置仍然包括了收发两端及交换机，交换机处于控制的核心位置，因为交换机从端口缓冲区占用率得到的控制信息，是平衡网络负载、降低网络拥塞扩散的关键。DeTail 能够降低拥塞扩散，降低传输时延，平衡网络负载，但对交换机改动太大，并不适合大规模部署。

Alizadeh M 于 2015 年提出 pFabric[41] ，它提供了一种兼顾短流和长流的基于优先级的解决方案，其实现基础是数据中心网络实际应用产生需要低时延的短流(用户请求等应用)和对时延不敏感的长流(备份、拷贝等应用)，降低时延的关键在于降低短流等待时间，如果优先传送短流，实现最早完成的流优先，就能够既降低短流时延，又不影响平均时延。因此，pFabric 将流调度和速率控制分开，流调度基于优先级，交换机只需要简单的严格按照优先级传送数据，就能实现最优的传送。当然，如果持续产生大量丢包，仍然会影响性能，所以速率控制的唯一目的就是在高持续丢包情况下降速，便可以达到接近理想状态下的最佳值。

对于流调度，pFabric 将整个数据中心网络看成一个大的交换机，如图 11 所示：

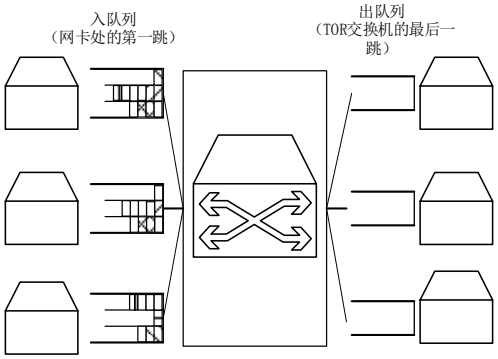


图 11：数据中心网络中的流调度示意图

如果将整个数据中心网络看成一个大的交换机，那么最佳的流调度策略就是总调度剩余量最少(优先级最高)，流交换机通过二叉树等数据结构找到优先级最高的流，使得整个网络流调度达到最佳。

pFabric 主要提出一种交换机的概念原型，这种交换机的队列，在入队列时要满时将当前队列中优先级最低的包丢弃；在出队列时，将当前优先级最高的包弹出。如果拥有这样的交换机，端节点只需要对所发送的包进行合适的优先级标识，数据中心中对于小流的时延就可以被认为是最优的。但是作者只考虑了概念模型，真实的交换机很难进行该操作。

Wei Bai 等人于 2015 年提出算法 PIAS[42] ，它是一个流调度算法，目的是达成短流高优先级，长流低优先级。许多现有数据中心网络（数据中心网络）流调度方案，基于假定的流量和交换机流量分

布先验知识实现流量完成时间最小化（FCT），使它们在性能上更优越，但在实践中很难实现。相比之下，PIAS 假定流分布是不可知的，从而寻求将 FCT 最小化。

该机制旨在通过模拟最短作业（SJF）来最小化 FCT，前提是不知道流量大小。从本质上讲，PIAS 利用现有交换机中可用的多个优先级队列来实现多级反馈队列（MLFQ），在这种队列中，PIAS 流会根据其发送字节数逐渐从高优先级队列降级为低优先级队列。因此，短流能在在前几个高优先级队列中完成，这使得 PIAS 能够在无法预先知道流量大小的情况下模拟 SJF。

如图 12 所示，PIAS 只部署在端节点，而不用在交换机上部署。问题的难点在于如何准确划分包长阈值 K，以及端与端之间的配合。阈值不准确和优先级之间失匹配，都会导致性能损失。文中，作者虽然通过建模给出了如何准确计算阈值和解决失匹配问题，但是仍然需要很长的时间使模型收敛。同时，它的优先级匹配的过程，存在慢启动的问题。即长流可能需要很长时间才能达到一个准确的优先级，导致性能损失。

PIAS 首次将不使用流大小的先验信息引入流调度，开创了新的方向，相比先前的研究，它的普适性更强。

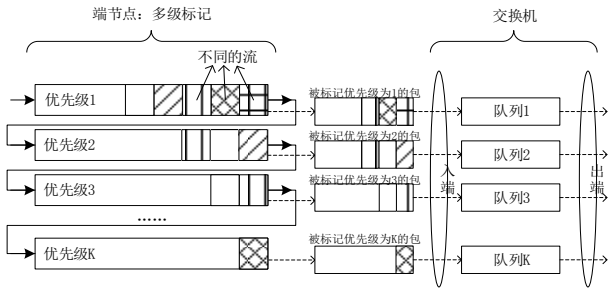


图 12：PIAS 结构

P. X. Gao 等人于 2015 年提出 pHost[43]，其目的是将 pFabric 和 Fastpass 的优势结合：pFabric 的最佳性能和 Fastpass 的商用网络设计。与 Fastpass 类似，pHost 通过将网络结构与调度决策分离来保持网络的简单性。pHost 引入了一种新的分布式控制算法，允许终端主机直接进行调度决策，从而避免了 Fastpass 集中式调度器架构的开销。

pHost 最先提出 Receiver-driven 的想法，将发送端的发送功能和速率控制分离，具体的，接收端先发送一个类似 ACK 包到发送端，发送端后发送一个数据包给接收端。pHost 使用了包级别的负载均衡机制 packet-spraying，当拥有多条等价路由时，它会将包随机转发到任意一条上。因此在数据中心中，核心层和汇聚层基本上不会发生丢包，丢包只发生在边缘层。Receiver-driven 认为由接收端来控制发送端的速率，将会避免接收端的拥塞问题。实验结果也证实了这一结论。同时，Receiver-driven 可以通过在接收端进行优先级调度，因为接收端知道到达它的每条流的信息，因此在接收端对回传的 ACK 进行优先级排序，就可以达成优先级调度的效果。

pHost 的问题是会出现包乱序和不稳定的情况。因为使用了包级别的负载均衡，则必然出现乱序和头包阻塞问题。同时 burst 的随机性加上包转发的随机性，可能会增加 Incast 的程度。但是 Receiver-driven 的提出开启了一个新的设计空间，分布式的控制是新方法设计的趋势。

M. Handley 等人于 2017 年提出 NDP[44]，其主要目标是短流的低完成时延，以及可预测的高吞吐量。为了完全满足这些目标，NDP 设计了整个网络传输行为，包括交换机行为、路由和一个全新的流控传输协议。

NDP 可以说是 Receiver-driven 的继任者，它将 Receiver-driven 的一些问题进一步完善。首先利用优先级调度，对不同优先级的包采用不同的调度策略，其次使用类似 CP（Cut Payload）[33] 的方法，即当发生拥塞时，将数据包的负载减掉，并且及时回传至发送端，从而达到更快的通知发送端减速以及回传的目的。最后利用 Receiver-driven 实现控制。

NDP 的优点是完善了 Receiver-driven，使之成为一个较为完善的系统。但是重新设计一个新的传输协议过于激进，很难真正商用。

Montazeri B 等人于 2018 年提出 Homa[45]。它提供了异常低的时延，特别是对于具有大量超短流的工作负载，并且它还支持长流和高网络利用率。Homa 使用网络优先级队列来确保短流的低时延；优先级分配由每个接收方动态管理，并与 Receiver-driven 的流量控制机制集成。Homa 还使用了受控的接收机下行链路，以确保高负载的有效带宽利用率。在仿真实验中，Homa 的时延大致相当于 pFabric，并且比 pHost、PIAS 和 NDP 的速度要好得多，几乎所有的流量大小和工作负载都是如此。Homa 也能承受比 pFabric、pHost 或 PIAS 更高的网络负载。

Homa 的出发点源于，小流需要更加精细的优先级控制，所以它设计了三个机制来保证小流优先。首先，更多的优先级控制；其次，发送端的 SRPT（最短剩余作业时间调度法）；最后，接收端的动态分配优先级机制。它也是 Receiver-driven 的代表，将 Receiver-driven 进一步完善。同时为了保证因为优先级产生的带宽浪费，它还设计了过度提交机制。

Homa 的优点是对优先级进行了优化，优先级调度变得更为精细。但是缺点也很明显，如何更好的分配优先级是一个困难的问题，文章中只是利用先验知识直接划分了优先级。同时 Homa 是一个无

连接的，不保存状态信息，但是包的完整性和连接的可靠性需要上层协议保证。

在数据中心中，流量优化（Traffic Optimization）是一种在线决策。在此之前，大多数研究使用启发式方法，需要依赖人对工作负载和环境的理解。因此，设计和实现算法至少需要几周时间。Li Chen 等人于 2018 年提出算法 AuTo[46]，它使用了深度强化学习（DRL， Deep Reinforcement Learning）技术在线解决流量优化问题。作者认为，简单的 DRL 并不能解决问题，因为对于数据中心中很多流量，在 DRL 作出决策之前，流已经完成了，DRL 失去了它的意义。

利用数据中心中流量的长尾分布，AuTo 开发了一个两级的 DRL 系统，它通过模仿动物的边缘神经系统和中枢神经系统的结构，以解决模型的灵活性。如图 13 所示，一个快速的决策模型，运行在端节点处理短流，利用 PIAS 的结构，但是 PIAS 的每一个阈值是利用一个 DRL 学习得到的，不一定实时更新。一个中心的大型 DRL，用来调度长流，同时进行路由的选择。AuTo 是一个端到端的自动系统，它可以收集网络信息，从过去的决策中学习，并执行操作来实现目标。

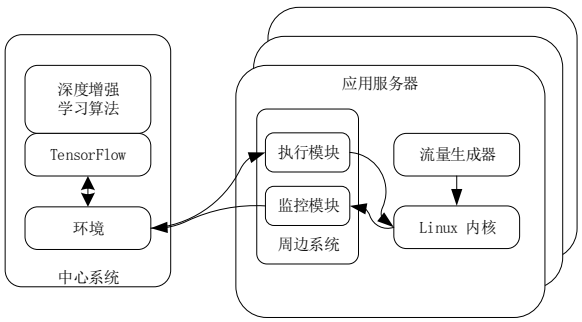


图 13: AuTo 结构示意图

AuTo 的开创性在于首次将机器学习和人工智能方法应用于传统系统设计。传统机器学习应用只是在应用层上面进行，没有设计基础的流量控制。然而 AuTo 的问题也十分明显，现阶段机器学习设备的处理和通信开销使得它无法实现系统大规模部署，从而难以投入实际商用。

3.3.1.3 基于链路价格的优先级调度

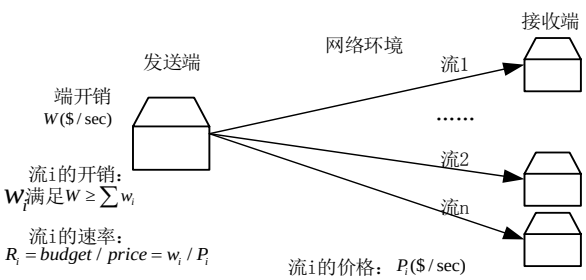


图 14:FCP 设计

Han D 等人于 2013 年提出 FCP (Flexible Control Protocol)[47]，其中认为数据中心网络中显式拥塞控制算法对交换机的过分依赖导致收发两端灵活性降低，削弱了 TCP 自带的在收发两端灵活使用不同算法的能力。在 FCP 里，交换机只是反映数据中心网络链路信息，最终的决策在发送端实现。FCP 通过聚合和本地控制实现灵活性，聚合确保一组流共享一个带宽，本地控制允许发送端自主决定每条流的带宽大小，从而实现较好的灵活性。FCP 的主要设计如图 14 所示。对每个发送端 h，确保单位时间内所有数据包(Packets)的发送总和小于预算，即满足式(7)：

$$\sum_{s \in Packets} price(s) \times size(s) \leq W_h \tag{7}$$

显然 FCP 需要处理的是流量剧烈变化的情况，FCP 通过发送端预加载，向网络通知发送端下一轮相对当前增加的预算倍数，达到快速收敛的目的。

FCP 增强了端节点的灵活性，但不论是预加载或链路定价反馈，对于发送端而言都具有滞后性，FCP 过于激进的使用了预算，在动态变化的数据中心网络中是否具有较好的性能，还需要进一步验证。

3.3.1.4 优先级调度总结

对流量进行优化的前提，就是需要给流量一个评级，如何定义合适的优先级成为一个研究问题。如表 2 所示。

表 2 不同的优先级

名字	优先级的依据	应用需求
PDQ[39], DeTail[40], pFabric[41], Fastpass[34], PIAS[42], pHost[43], NDP[44], Homa[45], AuTo[46]	流量大小	小流需要更短的流完成时间
D3[37], D2TCP[38], PDQ[39]	流量截止时间	截止时间越早越优先
FCP[47]	链路价格	综合数据包的重要性和长度

PDQ、Homa、AuTo 等一系列使用流量的大小作为优先级；D3、D2TCP 等使用 deadline 来确定优先级；FCP 使用链路的价格来确定优先级。

可以看到主流的优先级调度是依据流长短进行的。然而不同的算法对于流优先级的看法也不同、使用优先级的方法也不同。

表 3 流量分布是否可知

名字	优先级的依据	优缺点
AuTo[46], PIAS[42]	流量分布不可	更加友好，可

	知	直接部署
PDQ[39], DeTail[40], pFabric[41], Fastpass[34], Homa[45]	流量分布可知	需要修改上层协议，提供流量信息

流调度的很多算法都假定流的大小分布已知，大多数算法要求在流还未发送之前就需要知道它的长度以确定优先级。然而在很多情况下，这个要求很难实现。因此 PIAS 和 AuTo 假设流大小即分布不可知，进行优先级调度。流量分布是否可知总结如表 3 所示。因为有了流量不可知这一假设，使算法灵活性增加，从而使算法更容易部署。

众所周知，传统 TCP 的速度控制是基于发送端的，发送端一次可以发送一个窗口的包，当速率过快，就要减小窗口大小。然而这样的设计基础是广域网的结构复杂、不稳定、易丢包等情况。在数据中心中，结构相对简单，连接相对可靠，并且距离短、速度快，TCP 依靠端决策就有些局促，因此提出很多基于中心控制或基于接收端的算法，如表 4 所示。

驱动位置是指控制发送速率的控制权的位置。接收端驱动，需要包级别的负载均衡配合使得核心层的拥塞消除，从而使拥塞发生在最后一跳交换机上，因此接收端拥有比发送端更多的拥塞信息。

表 4 算法的驱动位置

名字	驱动位置	可部署性
TCP Cubic[28], DCTCP[5], DCQCN[30], HPCC[36]	发送端	可部署性高
NDP[44], Homa[45], pHost[43]	接收端驱动	定制硬件，可部署性低
ExpressPass[35]	Credit 驱动	专用硬件
FastPass[34]	中心化驱动器驱动	中心化硬件要求高

于是，将接收端的下行瓶颈带宽充分使用，即可避免整个网络的拥塞。中心化的驱动可以达到全局最优，但是中心的传输会耗费额外的带宽，并且大规模部署存在实现问题。

3.3.2 负载均衡

数据中心网络的拓扑一般为 CLOS 结构，主机之间经常存在多条等价的冗余路径。数据中心为了满足各种应用需求会提供大量带宽资源。负载均衡目的就是依据拓扑结构明确、路径资源冗余的特性，尽可能将流量在多条等价路径中均匀分布，避免网络拥塞，对网络资源进行充分的利用。

3.3.2.1 简单机制

ECMP[48] 全称等价多路径（Equal-cost multi-path），它是一种基于流的负载均衡路由策略，当路由器发现同一目的地址存在多条等价路径时，路由器会依据相应算法，将不同流量分布到不同的链路上，以增进网络带宽利用率。

ECMP 的路径选择策略有多种算法，哈希，例如利用流的五元组哈希为流选择路径；轮询，各个流在多条路径之间轮询选择；基于路径权重，根据权重系数，系数大的分配流量多。ECMP 是一种简单的负载均衡策略，在实际使用中存在非常多的问题。首先，它可能加重网络链路的拥塞。由于只是进行哈希或者轮询，ECMP 并不能感知拥塞，对于已经产生拥塞的链路来说，很可能加剧该链路的拥塞。其次，非对称网络的性能损失。当数据中心网络出现故障时，网络结构出现非对称，因此网络物理链路无法达到均衡分布，进而导致流量不均衡。最后，ECMP 在流量大小分布均匀的条件下效果好，然而在大象流和老鼠流并存的情况下，效果并不理想。假设一条大象流和一条老鼠流同时到达路由器，ECMP 将两条流平均分到了两条等价路径上，显然这时候等价的路径并没有被高效的利用。因此，将 ECMP 直接部署在数据中心这种突发流量多，大象流与老鼠流并存的环境中，需要仔细考虑环境的问题。

Packet Spraying[49] 也称 Random Packet Spraying (RPS)，中文翻译为随机包喷洒。它是一种基于包级别的负载均衡策略，当路由器发现同一目的地址存在多条等价路径，将会以包为单位分布到各个链路上。与 ECMP 不同的是，RPS 是以包为单位的，ECMP 是以流为单位的，RPS 会将同一条流的不同包转发到不同的等价路径上。

RPS 的好处是简单，易实现，并且可以将充分利用网络链路，不会存在突发流或长短流出现的情况下，网络出现不均衡的情况。然而 RPS 并不是没有任何缺点，对于 TCP 来说，RPS 会导致严重的失序问题。比如，最后发出的包可能最先到达，因为 TCP 无法分辨包的失序和丢包情况，它将会出发拥塞避免机制，拥塞窗口减半，从而导致效率降低。RPS 可以保证核心层交换机处于不拥塞状态，而会造成汇聚层严重拥塞，如果没有准确的拥塞控制机制保证 RPS 汇聚层拥塞的问题，它仍会影响性能。

当前 RPS 主要和其他拥塞控制或者流调度配合使用，如与 pHost、NDP 和 Homa 等一类 Receiver-driven 的算法配合使用，Receiver-driven 可以让最后一跳交换机到接收端的下行链路不拥塞，从而一定程度上弥补了 RPS 的缺陷。

因此和 ECMP 类似，将 RPS 部署到数据中心中时，需要考虑各种环境因素，并且需要其它机制和技术与之配合。

3.3.2.2 复杂系统

随着研究深入，单个简单机制已经不能完整的

解决因为负载均衡而产生的各种问题，因此更多拥有拥塞感知，更加灵活的粒度控制，更加稳定的复杂系统出现了。

Al-Fares 等人于 2010 年提出 Hedera[50]，它是一种基于 SDN（Software Defined Network，软件定义网络）的负载均衡系统。它通过动态预测大象流的带宽，利用模拟退火算法和全局首次适应算法对流量进行分配，控制位置在 SDN 的控制器中。Hedera 的优势在于，利用 SDN 可以对网络有全局的管控；Hedera 的局限性在于，负载计算开销非常大，权衡性能，只能针对大流进行调度。

M. Alizadeh 等人于 2015 年提出 CONGA (Distributed Congestion-Aware Load Balancing)[51]，它是一种基于网络的分布式拥塞感知负载平衡系统，其设计目标是在不增加传输层复杂度的前提下，通过分布式方式实现全局负载平衡。控制位置主要在最后一跳交换机(leaf switches)，源交换机根据拥塞信息做出负载平衡决策。

CONGA 基于数据中心网络的特点，将流进一步细分为间隔粒度在微秒级别的小流(flowlets)，负载均衡也针对每一个 flowlet 的第一个包，之后每个 flowlet 使用相同的链路。上行链路交换机搜集链路拥塞状况并交给收端交换机，保存一个来自各叶节点的拥塞状况，并反馈给源端交换机。

CONGA 通过负载均衡提升了数据中心网络传输性能进而提高吞吐量，但 CONGA 仍然需要网络负载与实际容量相匹配，当实际容量无法满足时，CONGA 的性能无法得到保证。

K. He 等人于 2015 年提出 Presto[52]，它是一个基于端节点的分布式负载均衡系统。它将负载均衡的功能推入到网络的边缘，比如虚拟机中，这样就不需要在传输层和各种硬件中进行配置。其次 Presto 使用了近乎均匀分布的粒度单元（flowcells），然后为 cell 分配可用路径，这意味着划分为同一 cell 的短流可以保持其完整性。

与 CONGA 相比 Presto 的提升在于并未使用特定交换机，因而更适合一般数据中心升级。然而由于只是在端节点控制，很难有负载感知，很难控制大规模网络失效导致的性能损失。

S.Ghorbani 等人于 2017 年提出 DRILL[53]，首次在数据中心树形结构中使用微负载均衡，具体是指在微秒级别上尽可能实现均匀的分配负载。DRILL 依据交换机发送端口的队列长度来确定数据包转发路径，将数据包发送到排队队列长度最短的端口。由于交换机各端口的队列长度较为平均，因此每对相同源地址和目的地址之间的路径时延不会相差很大。与之前的机制相比，DRILL 使用了交换机局部拥塞信息来进行负载均衡，它也需要对交换机做一定程度的修改。

Zhang H 等人于 2017 年提出 Hermes[54]，它定义了由频繁重路由而导致的拥塞失配现象，并指出这种现象会导致传输性能的降低。拥塞失配是指拥塞控制由于重路由导致解决拥塞的位置与实际拥塞的位置不匹配的现象。因此，需要考虑这些方案的稳定性，尤其是负载增加时，许多数据流会在网络中相互作用，而频繁改变路由，出现拥塞失配现象，降低负载均衡的性能。Hermes 利用终端主机通过对 ECN 标志和链路时延的检测来感知链路故障，将频繁的超时和重传视为链路故障的标志。同时在终端主机对之间周期性发送探测数据包来提高对链路状态的可视性。

3.3.2.3 负载均衡总结

将流量均匀的分布到多条等价路径上，更好的利用网络冗余结构是负载均衡的目标。然而，采用流级别的 ECMP 并不均衡，采用包级别的 RPS 又会导致包的乱序。如何在二者之前权衡，或者通过其他方面补充完善负载均衡，是一个重要问题。

表 5 主流负载均衡粒度

名字	控制粒度	优缺点
ECMP[48] Hedera[50]	流级别	大小流均衡效果差
RPS[49] DRILL[53] Hermes[54]	包级别	均衡效果好，但是易出现乱序
CONGA[51] Presto[52]	Flowlet，Flowcell	介于包与流之间，存在权衡

表 5 总结了当前负载均衡机制的控制粒度。可以看到，为了达到更好的负载均衡效果，DRILL 和 Hermes 等仍然使用包级别的负载均衡，首先达到一个较好的负载均衡状态，再去解决包乱序的问题。介于包和流级别的 Flowlet 和 Flowcell 是一种好的思路，但是如何确定更加合适的粒度大小仍是未来的研究方向。

表 6 主流负载均衡感知程度

名字	感知程度	优缺点
ECMP[48], RPS[49] Presto[52]	无	没有感知，均衡效果差
CONGA[51]	全局感知拥塞（交换机）	中心调度瓶颈
DRILL[53]	局部拥塞感知（交换机）	分布式感知存在协同问题
Hermes[54]	全局拥塞感知（端节点） 交换机故障感知	易于部署，需要与拥塞控制配合

表 6 对比了各个负载均衡方法对于拥塞感知的

状况。拥塞对于负载均衡的性能影响是不可忽视的，因此如何更加准确的感知拥塞，并且与相关负载均衡机制相结合也是当前的研究重点。

3.4 数据中心流量控制总结与综合比较

表 7 综合对比了近年来研究人员所提出的数据中心的流量控制算法或者技术。通过每个算法或者技术的突出特点可以看到，性能提升是通过获取更多的相关信息，或通过其他相关资源互换实现。DCTCP、DCQCN 通过 ECN 的通知来获得拥塞信息，从而做出决策；HPCC 通过更加准确的 INT 来获取拥塞信息；TIMELY 是通过 RTT 的变化获得拥塞信息的；D3 和 D2TCP 是通过 deadline 来对流进行优先级的划分；Homa，NDP，Fastpass，pFabric 等都不同程度的提出为短流提供更高的优先级以保证传输性能。

拥塞控制、流调度和负载均衡这三个主要的研究点，主流算法系统只会着重设计其中的一至两个。如 DCTCP、DCQCN、CONGA 等，只是涉及一个主要的方面；D3、D2TCP、pFabric、NDP、Homa 等虽然涉及三个方面，但是主要是拥塞控制和流调度，只是使用了简单的负载均衡机制来对协议进行补充。因此如何综合设计流量控制的各个部分，使之成为一个整体，相互促进，显得十分重要。

各流量控制算法是否需要专用硬件一定程度上反映了该设计方案能否快速部署到实际应用中。但是当前只有少数如 Presto、PIAS、Hermes 等不需要专用的设备，可以直接快速部署在当前数据中心以外，其他算法机制都无法大规模部署。

除了 DCTCP 和 DCQCN 等拥有已商用的硬件外，其他都需要定制化的硬件，这些算法可能需要经历很长时间才能实现实际部署和应用；而 DCTCP 和 DCQCN，尽管这些算法已经具备技术的可实现行，但是相关硬件的成本会限制这类算法技术的实际部署。

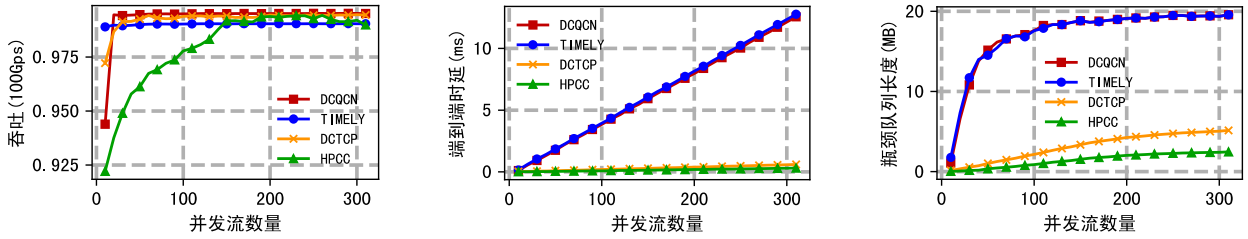
4 仿真实验对比

由于拥塞控制已经大规模部署在商用数据中心中，并且相关算法已经开源，因此本节选择四种拥塞控制机制进行 NS-3 仿真对比，它们分别是 DCTCP、DCQCN、TIMELY 和 HPCC。通过两个典型的数据中心流量模式 Incast 和真实流量模式，对四种算法进行性能上的分析比较。

表 7 数据中心流量控制的综合比较

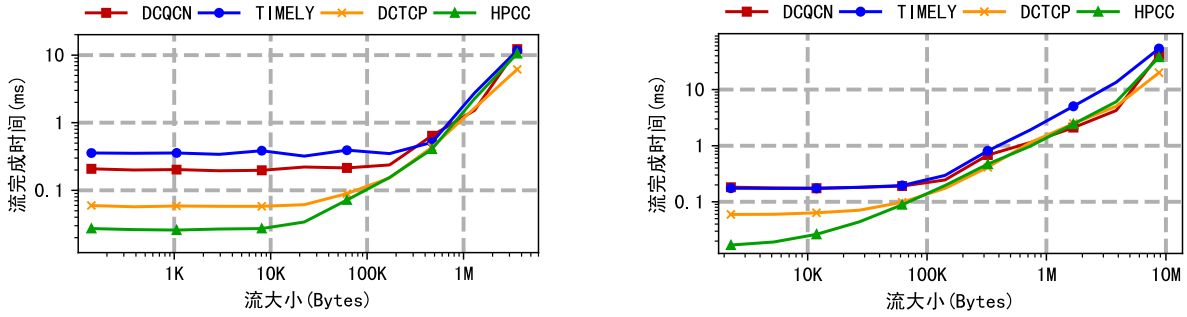
名字	突出特点	主要类型	是否需要专门硬件
DCTCP[5]	开创了数据中心传输层拥塞控制，利用 ECN 进行高效反馈	拥塞控制	已商用
Hedera[50]	使用 SDN 对数据中心的大象流进行调度	负载均衡	SDN 设备
D3[37]	基于 deadline 实现各类流的差分服务	流调度、拥塞控制、ECMP	是
D2TCP[38]	DCTCP 和 D3 的结合	流调度、拥塞控制、ECMP	是
PDQ[39]	实现全局范围内的 EDF 和 SJF	流调度、拥塞控制、ECMP	是
HULL[29]	提前发现拥塞和为小流预留传输带宽	拥塞控制、ECMP	是
DeTail[40]	通过跨层配合减少长尾，对短流优先级排序，平衡网络负载	流调度、拥塞控制、RPS	是
FCP[47]	发送端根据交换机反馈信息实现拥塞控制	流调度、拥塞控制	是
pFabric[41]	兼顾短流和长流的基于优先级的解决方案	流调度、拥塞控制、RPS	是
Fastpass[34]	集中式的调度实现全局最优	中心化调度	是
CONGA[51]	通过分布式方式实现全局负载平衡	负载均衡	是
TIMELY[32]	基于 RTT 测量的拥塞控制	拥塞控制	是
DCQCN[30]	基于 RoCEv2，是对 RDMA 拥塞控制的开创	拥塞控制	已商用
Presto[52]	将负载均衡做到端节点	负载均衡	否
PIAS[42]	流长度不可知性的流调度算法	流调度	否
pHost[43]	Receiver-driven	流调度、拥塞控制、RPS	否
NDP[44]	短流低延迟，使拥塞信息更快通知	拥塞控制、流调度、RPS	是
ExpressPass[35]	基于 credit 的拥塞控制	拥塞控制、ECMP	是

DRILL[53]	利用交换机队列长度做负载均衡	负载均衡	是
Hermes[54]	解决频繁重路由导致的拥塞	负载均衡	否
Homa[45]	短流需要更精细的调度	流调度、拥塞控制、RPS	是
AuTo[46]	利用 DRL 进行流调度	流调度	是
HPCC[36]	利用 INT 进行拥塞感知	拥塞控制	可商用



（a）吞吐　　（b）时延　　（c）瓶颈队列长度

图 15 Incast 场景下的吞吐、时延、瓶颈队列长度随并发流数量变化图



（a）FBHadoop 流量分布 (70%平均负载)　　（b）WebSearch 流量分布(70%平均负载)

图 16 真实场景下的 99th 流完成时间随流大小变化图

实验拓扑为 FatTree，有 16 个核心交换机、20 个汇聚层交换机、20 个 ToR 交换机和 320 个服务器（每个机架上有 16 个）。每个服务器都有一个 100Gbps 的网卡连接到一个 ToR 交换机上。核心交换机与汇聚层交换机之间、汇聚层交换机和 ToR 交换机之间的每个链路的容量均为 400Gbps。所有的链路都有 1us 的传播延迟，也就是说最大的 RTT 为 12us。交换机为共享缓存交换机，其缓存大小为 32MB。相关代码源自文献[36] 。

4.1 Incast场景

在 Incast 场景中，从 320 个主机中任意选出 N 个主机向一个固定的主机发送一条大小为 100KB 的数据流。并发主机数量 N 从 10 到 320 变化。图 15 展示了实验结果。

首先在吞吐方面，可以看到 DCTCP、DCQCN 和 TIMELY 都可以有效的利用瓶颈链路的吞吐，但是 HPCC 在并发数量小于 150 时，并不能保证完全利用吞吐，原因在于 HPCC 需要预留 5%的吞吐来提供低时延。

在时延方面，可以看到 DCQCN 和 TIMELY 随着并发流量数目的增加，端到端时延增长快。而相对的 DCTCP 和 HPCC 都可以提供超低的时延，并且 HPCC 比 DCTCP 更低。在这里 DCTCP 优于 DCQCN 的原因在于本实验的仿真均来自算法层面，而不包括软件和硬件的差异造成的差异。因为 DCTCP 不能绕过内核，所以存在显著的软件级和硬件时延[30] 。

在队列长度方面，HPCC 同样展现出好的性能，随着并发数量的增加，队列长度增长不明显。

4.2 真实流量模式

在真实流量模式中，使用了被广泛接受和公开的数据中心流量分布 WebSearch [30] 和 FBHadoop[12] 。具体的，从 320 个服务器中选取前 60 个服务器，每个服务器向其他 59 个服务器发送 1000 条数据流，目标服务器随机，流量大小服从真实的数据中心流量分布，流量的发出时间服从负指数分布，使得流量的发送为泊松流，其中参数的计算方法为，根据所需要的负载大小（如 70%），算出平均每条数据流的发送间隔时间，间隔时间就是负指数分布的参数。

实验结果如图 16 所示，可以看出，无论是超过 90%低于 120KB 的 FBHadoop 数据分布，还是大流占比相对较多的 WebSearch 数据分布，HPCC 对于小流都是十分友好的。图 16（a）和 16（b）显示 99th 的小流完成时间，HPCC 对于短流可以达到

非常低的流完成时间，20us 之下。仅仅比基础 RTT 时延 12us 多了 8us。HPCC 这种效果符合数据中心的应用需求，小流的应用一般是时延敏感的，而大流一般是吞吐敏感的。

4.3 仿真总结

综合两个场景，可以看出，HPCC 是目前最符合数据中心应用场景的拥塞控制算法，小流的低时延和大流的相对高吞吐。但是，这个结论只能从算法层面得出。从成本角度，DCTCP 使用传统的 TCP/IP 协议栈，利用传统网卡即可；DCQCN 和 TIMELY 需要使用定制的 RDMA 网卡，和支持无损网络的交换机等等；而 HPCC 除了以上以外，仍需要支持 INT 的网卡和交换机。因此，高效利用目前的设备以提供低时延，或者支持可增量的硬件部署也是未来研究热点之一。

5 未来研究方向

数据中心网络体系结构属于互联网体系结构中的演进式[55]，但是相比于传统网络架构，它可以更加革命，可以强调全新的设计，尤其是全新的流量控制技术。与传统流量控制相比，研究人员对于数据中心网络传输技术的研究处于百家争鸣的状态，蓬勃发展但尚不充分。这主要体现在：（1）多数方案仅处于原型验证阶段，硬件定制化程度高，难以将众多算法放在同一个环境中进行对比。并且方案缺少低成本、成熟的商用设备支持；（2）数据中心中的大量应用仍未被开发出来，机器学习人工智能等平台的分布式应用仍未大量部署，目前仅仅作为云计算的基础设施平台；（3）工业界的主要通讯设备提供商，华为、中兴、思科等，研发数据中心设备较少，国际标准化组织如 IEEE、IETF 等针对数据中心提出的技术标准或者建议仍有待进一步地丰富和完善。

因此数据中心传输技术还需要进行更加广泛深入的研究，未来研究方向将集中于以下几个方面：

（1）统一的流量控制测试平台

当前已经提出了许多数据中心网络流量控制的算法，但是哪种最好难以评判，要放在一起对比测试非常困难。数据中心不像广域网，数据中心的软硬件定制化程度是比较高的。因此，无法把多个算法放在一个数据中心里进行对比。但在广域网则要容易得多，例如 Pantheon[56] 平台，就是专门对广域网拥塞控制协议进行对比的评估平台。因此，一个新的研究方向是开发一个统一通用的流量控制测试平台，研发人员只需提供算法就可以进行统一评价，综合对比各个算法，各种参数下的性能，为后续研究提供便利。

（2）综合考虑拥塞控制、流调度和负载均衡的研究

流量控制的研究点就在于拥塞控制、流调度和负载均衡三个具体方面，他们的总体目标都是为了给用户提供更好的体验，然而三个方面的目标各有差异。甚至，在进行机制和算法的设计过程中，三个方面会相互冲突。如 3.4 节所述，当前研究主要着力于其中的一两点，尚未存在一个传输技术可以综合考虑三方面。因此，未来对于如何将拥塞控制、流调度和负载均衡综合设计会成为重点，非冲突的、互补的各种机制方法组合起来的传输技术会极大的改善数据中心传输性能。

（3）中心化控制和分布式控制的发展方向

在广域网中，软件定义网络（Software Define Networking，SDN）因为控制和转发分离等特性，被工业界学术界应用研究，如 Google[57]、微软[58]、以及国内外高校[59] 等等。在数据中心内部，相比于分布式，集中式控制可以对网络全局信息进行掌控，更能精确的规划控制流量[60] [61]。然而 SDN 的中心化会导致效率和可部署问题，如 Hedera 无法规划小流，只能针对大流进行规划；FastPass 中心化面临可部署规模问题，当数据中心规模增加，中心化的控制器成为网络瓶颈。而传统的基于发送端的分布式流量控制，由于没有足够的信息，无法准确调度。因为数据中心特殊的应用模式，接收端拥有更多的流量信息[44] [45] [62]，接收端驱动是一个新的研究方向。一些新的研究也在交换机上着力，认为交换机可以提供除了转发以外更多的控制能力，如 ABQ[63]，一个主动缓冲反向 ACK 以配合端节点拥塞控制的交换机机制；SP-PIFO[64]，通过严格优先级队列实现 PIFO 的任意优先级队列。因此，如何将 SDN、发送端驱动、接收端驱动、交换机驱动等一系列控制方法有效的应用到流量控制中是一个重要研究方向。

（4）高性能专用设备的研发部署使用

网络硬件技术的提升，对于网络传输性能的提升是巨大的。目前，基于 RDMA 的传输技术已经成为了工业界和学术界热点[65]。RDMA 通过使客户端可以直接访问服务器的内存而不需经过服务器 CPU 耗时的传输，减小通讯对 CPU 的使用，以提升传输性能。同时，可编程的网络设备极大程度的提升网络的灵活性和快速部署能力。如 SDN 设备，支持 P4[66] 编程语言的交换机，NetFPGA[67] 智能网卡等等。目前学术界对于原型系统的设计实现主要依靠这些可编程网络设备，如 Hedera 使用 SDN 实现，NDP 使用 NetFPGA 和 P4 交换机实现。未来，光互联技术将成为数据中心重要的研究方向[11]。光互联技术由于其高带宽，高容量，低开销，低能耗等特性，可以有效提升网络传输的性能。目前为了有效兼容现有的电互联技术，国内外研究提出光电

混合结构[68] [69] 。如何应用好当前的高性能设备，如何利用新型设备，如何开发新的高性能专用设备都将成为未来的研究热点。

（5）适合特定应用场景的流量控制研究

算法性能的提升是通过获取更多的相关信息，或是通过其他相关资源互换得来的。在特定应用场景下获得的资源就更多。例如 Zhang Yuchao 等人提出的 D3G[3] 框架，它针对在云服务器中的链式服务，根据应用的特征，有针对性的进行网络调度优化，以减小服务的整体排队时延。Daehyeok K 等人，针对大规模的数据存储数据应用场景，设计 HyperLoop[70] 系统，它使用 RDMA 技术，针对网络分布式存储过程中，数据流传输的过程和特性，将传输的过程充分卸载至高速智能网卡（SmartNIC），通过节约 CPU 时间，以达到网络传输的高效性。Kim D 等人提出一个融合容器和 RDMA 的“OvS”系统 FreeFlow[71] 。与传统 OvS 系统不同的是，FreeFlow 通过分析 OvS 的数据传输特性，根据应用服务提供了 API 级别的控制，同时使用 RDMA 提升性能。

如何更好更充分的利用这些信息，或者依照特定的应用场景获取更多的信息进行设计算法。为开发新的交换机硬件、智能网卡（SmartNIC）提供思路，是未来的研究方向。

（6）人工智能与流量控制技术结合的研究

人工智能以其强大的自适应性、自学习能力为各个研究领域提供了一套有效的决策工具，近年来引起了各界的广泛关注和研究。如何将人工智能技术与网络传输技术相结合也是未来研究的热点。在广域网中，PCC[72] 、PCC Vivace[73] 、Indigo[56] 等都试图将人工智能算法应用于拥塞控制，但是在数据中心中，由于数据传输的速率远远大于人工智能算法的决策速度，因此很有可能导致决策还没有发出，流量已经传输完成的情况；在流媒体传输中，Pensieve[74] 使用 DRL 对网络传输过程中的码率进行调整，以达到对应的 QoS；AuTo 是首个将 DRL 应用于数据中心流调度问题中的算法，通过在线学习模型，一定程度上解决了学习模型的效率问题。可以预料，未来如何解决学习模型的效率、如何将人工智能算法与传输技术相结合，这一方向将产生更多的研究成果。

6 结论

随着 5G 时代的到来，随着机器学习、人工智能、VR、AR 的发展，数据中心成为网络中的数据存储和计算的重要纽带。国内外学术界、国际标准化组织、网络设备提供商、云计算平台等都对数据中心网络的研究给与了非常大的关注。

因为数据中心领域的研究与工业界联系十分紧密，设备、技术、协议上的创新易于部署，性能的提升显而易见，未来关于数据中心网络的研究将会持续成为焦点。数据中心网络流量控制的研究将会成为网络发展的助推器。

尽管该领域仍存在一系列待解决的问题，但可以预期的是，随着各种新网络硬件的开发和研究的深入，低成本、高带宽、低时延的数据中心网络流量控制将有力推动云计算技术的发展。

致 谢 衷心感谢评审专家和编辑们对本文提出的宝贵意见和建议！

参 考 文 献

[1] Greenberg A , Hamilton J R , Jain N , et al. VL2: A scalable and flexible data center network//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Barcelona, Spain, 2009: 95-104

[2] Guo C . DCell: A scalable and fault-tolerant network structure for data centers//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Seattle, USA, 2008: 75-86

[3] Zhang Yuchao, Ke Xu, Haiyang Wang, Qi Li, Tong Li, and Xuan Cao. Going fast and fair: Latency optimization for cloud-based service chains. IEEE Network, 2017, 32(2): 138-143

[4] Al-Fares M , Loukissas A , Vahdat A . A scalable, commodity data center network architecture//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Seattle, USA, 2008: 63-74

[5] Alizadeh M, Greenberg A, Maltz D A, et al. Data center tcp (dctcp)//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). New Delhi, India, 2010: 63-74

[6] Shen M, Liu H, Xu K, Wang N, Zhong Y. Routing on demand: toward the energy-aware traffic engineering with OSPF//Proceedings of the International Conference on Research in Networking. Berlin, Germany, 2012: 232-246.

[7] Ke Xu, Meng Shen, Hongying Liu, Jiangchuan Liu, Fan Li, and Tong Li. Achieving Optimal Traffic Engineering Using a Generalized Routing Framework. IEEE Transactions on Parallel and Distributed Systems (TPDS), 2016, 27(1): 51-65.

[8] Meng Shen, Mingwei Wei, Liehuang Zhu, et al. Classification of Encrypted Traffic with Second-Order Markov Chains and application Attribute Bigrams. IEEE Transactions. Information Forensics and Security, 2017, 12(8): 1830-1843.

[9] Zhang Y, Ansari N. On architecture design, congestion notification, TCP incast and power consumption in data centers. IEEE Communication Surveys & Tutorials, 2013, 15(1): 39-64

[10] Li Dan, Chen Gui-Hai, Ren Feng-Yuan, et al. Data center network research progress and trends. Chinese Journal of Computers, 2014 37(2): 259-274(in Chinese)
(李丹, 陈贵海, 任丰原等. 数据中心网络的研究进展与趋势. 计算机学报, 2014, 37(2): 259-274)

[11] Yu Xiao-Shan, Wang Kun, Gu Hua-Xi, et al. The optical interconnection network for cloud computing data centers: State of the art and future research. Chinese Journal of Computers, 2015, 38(10):1924-1945 (in Chinese)
(余晓杉, 王琨, 顾华玺等. 云计算数据中心光互连网络:研究现状与趋势. 计算机学报, 2015, 38(10):1924-1945)

[12] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren. Inside the social network's (Datacenter) network//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). London, UK, 2015:123-137

[13] Dean J , Ghemawat S . MapReduce: Simplified data processing on large clusters//Proceedings of the Symposium on Operating System Design and Implementation (OSDI). San Francisco, USA, 2004: 137–150

[14] Tang H , Gulbeden A , Zhou J , et al. The panasas active scale storage cluster - Delivering Scalable High Bandwidth Storage//Proceedings of the International Conference on Supercomputing (ISC). Pittsburgh, USA, 2004:53-63

[15] Kandula S , Sengupta S , Greenberg A G , et al. The nature of data center traffic: measurements & analysis//Proceedings of the Acm Sigcomm Conference on Internet Measurement Conference (IMC). Chicago, USA, 2009: 202-208

[16] Theophilus Benson, Aditya Akella, and David Maltz. Network traffic characteristics of data centers in the wild//Proceedings of the ACM Sigcomm Conference on Internet Measurement Conference (IMC).Melbourne, Australia, 2010: 267-280

[17] Chowdhury M , Stoica I . Coflow: a networking abstraction for cluster applications//Proceedings of the ACM Workshop on Hot Topics in Networks (HotNets). New York, USA, 2012: 31-36

[18] Condie, T., Conway, N., Alvaro, P., Hellerstein, J. M., Elmeleegy, K., & Sears, R. MapReduce online//Proceedings of the Usenix Conference on Networked Systems Design and Implementation (NSDI). California, USA, 2010: 21-36

[19] Zhang, Jiaxing, Hucheng Zhou, Rishan Chen, Xuepeng Fan, Zhenyu Guo, Haoxiang Lin, Jack Y. Li, Wei Lin, Jingren Zhou, and Lidong Zhou. Optimizing data shuffling in data-parallel computation by understanding user-defined functions//Proceedings of the Usenix Conference on Networked Systems Design and Implementation (NSDI). California, USA, 2012: 295-308

[20] Mystore R, Pamboris A, Farrington N, et al. PortLand: A scalable fault-tolerant layer 2 data center network fabric//Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM). Barcelona, Spain, 2009: 39-50

[21] Ghemawat S , Gobioff H , Leung S T. The Google file system//Proceedings of the nineteenth ACM symposium on Operating systems principles (SOSP). New York, USA, 2003: 29-43

[22] Shvachko K , Kuang H , Radia S , et al. The Hadoop distributed file system//Proceedings of the IEEE Symposium on Mass Storage Systems and Technologies (MSST). Nevada, USA, 2010: 1-10

[23] IEEE. 802.11Qbb. Priority based flow control, 2011.

[24] IEEE 802.1Qaz. Enhanced Transmission Selection, 2011.

[25] IEEE. 802.11Qau. Congestion notification, 2010.

[26] IEEE 802.1AB. Station and Media Access Control Connectivity Discovery, 2011.

[27] Infiniband Trade Association. Supplement to InfiniBand architecture specification volume 1 release 1.2.2 annex A17: RoCEv2 (IP routable RoCE), 2014.

[28] Ha, S., Rhee, I. and Xu, L. CUBIC: a new TCP-friendly high-speed TCP variant//Proceedings of the ACM Special interest Group in Operating Systems (SIGOPS). New York, USA, 2008: 64-74

[29] Alizadeh M, Kabbani A, Edsall T, et al. Less is more: trading a little bandwidth for ultra-low latency in the data center//Proceedings of the Usenix Conference on Networked Systems Design and Implementation (NSDI). California, USA, 2012: 253-266

[30] Zhu Y, Eran H, Firestone D, Guo C, Lipshteyn M, Liron Y, Padhye J, Raindel S, Yahia MH, Zhang M. Congestion control for large-scale RDMA deployments//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). London, UK, 2015: 523-536

[31] Dragojević, A., Narayanan, D., Castro, M. and Hodson, O. FaRM: Fast remote memory//Proceedings of the Usenix Conference on Networked Systems Design and Implementation (NSDI). WA, USA, 2014: 401-414.

[32] R. Mittal, V. T. Lam, N. Dukkipati, E. Blem, H.Wassel, M. Ghobadi, A. Vahdat, Y. Wang, D. Wetherall, and D. Zats. TIMELY: RTT-based congestion control for the datacenter//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). London, UK, 2015: 537–550

[33] Cheng P, Ren F, Shu R, Lin C. Catch the whole lot in an action: Rapid precise packet loss notification in data center//Proceedings of the Usenix Conference on Networked Systems Design and Implementation (NSDI). WA, USA, 2014: 17-28.

[34] J. Perry, A. Ousterhout, H. Balakrishnan, D. Shah, and H. Fugal. Fastpass: A centralized "zero-queue" datacenter

network//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Chicago, USA, 2014: 307–318

[35] I. Cho, K. Jang, and D. Han. Credit-scheduled delay-bounded congestion control for datacenters//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). CA, USA, 2017: 239–252

[36] Li Y, Miao R, Liu HH, Zhuang Y, Feng F, Tang L, Cao Z, Zhang M, Kelly F, Alizadeh M, Yu M. HPCC: high precision congestion control//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Beijing, China, 2019: 44-58

[37] Wilson, C., Ballani, H., Karagiannis, T. and Rowtron, A. Better never than late: Meeting deadlines in datacenter networks//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Toronto, Canada, 2011: 50-61

[38] Vamanan, B., Hasan, J., & Vijaykumar, T. N. Deadline-aware datacenter tcp (d2tcp) //Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Helsinki, Finland, 2012: 115-126

[39] C.-Y. Hong, M. Caesar, and P. B. Godfrey. Finishing flows quickly with preemptive scheduling//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Helsinki, Finland, 2012: 127-138

[40] D. Zats et al. DeTail: Reducing the flow completion time tail in datacenter networks//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Helsinki, Finland, 2012: 139-150

[41] Alizadeh M, Yang S, Sharif M, et al. pFabric: minimal near-optimal datacenter transport//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Hong Kong, China, 2013: 435-446

[42] Wei Bai, Li Chen, Kai Chen, Dongsu Han, Chen Tian, and Weicheng Sun. Information-agnostic flow scheduling for data center networks//Proceedings of the Usenix Conference on Networked Systems Design and Implementation (NSDI). California, USA, 2015: 455-468

[43] P. X. Gao, A. Narayan, G. Kumar, R. Agarwal, S. Ratnasamy, and S. Shenker. pHost: Distributed near-optimal datacenter transport over commodity network fabric//Proceedings of the ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT). Heidelberg, Germany, 2015: 1-12

[44] M. Handley, C. Raiciu, A. Agache, A. Voinescu, A. W. Moore, G. Antichik, and M. Mojcik. Rearchitecting datacenter networks and stacks for low latency and high performance//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). CA, USA, 2017: 29-42

[45] Montazeri, B., Li, Y., Alizadeh, M., & Ousterhout, J. Homa: A receiver-driven low-latency transport protocol using network priorities//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Budapest, Hungary, 2018: 221-235

[46] Li Chen, Justinas Lingys, Kai Chen, and Feng Liu. AuTO: scaling deep reinforcement learning for datacenter-scale automatic traffic optimization//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Budapest, Hungary, 2018: 191-205

[47] Han, D., Grandl, R., Akella, A. and Seshan, S. FCP: a flexible transport framework for accommodating diversity//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Hong Kong, China, 2013: 135-146

[48] C. E. Hopps, Analsis of an equal-cost multi-path algorithm. InternetEng. Task Force, Fremont, CA, USA, RFC 2992, 2000.

[49] A. Dixit, P. Prakash, Y. C. Hu, and R. R. Kompella. On the impact of packet spraying in data center networks//Proceedings of the IEEE International Conference on Computer Communications (INFOCOM). Turin, Italy, 2013: 2130-2138

[50] Al-Fares, Mohammad, Sivasankar Radhakrishnan, Barath Raghavan, Nelson Huang, and Amin Vahdat. Hedera: dynamic flow scheduling for data center networks//Proceedings of the Usenix Conference on Networked Systems Design and Implementation (NSDI). CA, USA, 2010: 19-34

[51] M. Alizadeh, T. Edsall, S. Dharmapurikar, R. Vaidyanathan, K. Chu, A. Fingerhut, V. T. Lam, F. Matus, R. Pan, N. Yadav, and G. Varghese. CONGA: Distributed congestion-aware load balancing for datacenters//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Chicago, USA, 2014: 503–514

[52] K. He et al, Presto: Edge-based load balancing for fast datacenter networks//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). London, UK, 2015: 465-478

[53] S.Ghorbani, Z. Yang, P. B. Godfrey, Y. Ganjali, and A.Firoozshahian. DRILL: Micro load balancing for low-latency data center networks//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). CA, USA, 2017: 225-238

[54] Zhang H , Zhang J , Bai W , et al. Resilient datacenter load balancing in the wild//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). CA, USA, 2017: 253-266

[55] Xu Ke, Zhu Min, Lin Chuang. Internet architecture evaluation models，mechanisms，and methods. Chinese Journal of Computers, 2012, 35(10):1985-2006(in Chinses)
(徐恪，朱敏，林闯. 互联网体系结构评估模型、机制及方法研究综述. 计算机学报, 2012, 35(10):1985-2006)

[56] Yan, F.Y., Ma, J., Hill, G.D., Raghavan, D., Wahby, R.S., Levis, P. and Winstein, K. Pantheon: the training ground for Internet

congestion-control research//Proceedings of the USENIX Annual Technical Conference (ATC). MA, USA, 2018: 731-743

[57] Jain S, Kumar A, Mandal S, et al. B4: Experience with a globally-deployed software defined WAN//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Hong Kong, China, 2013: 3-14

[58] Hong C Y, Kandula S, Mahajan R, et al. Achieving high utilization with software-driven WAN//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Hong Kong, China, 2013: 15-26

[59] Wan Kao, Luo Xue-Feng, Jiang Yong, Xu Ke, et al. The Flow-oriented Scheduling Algorithms In SDN System. Chinese Journal of Computers, 2016, 39(6): 1028-1223(in Chinese)
(宛考, 罗雪峰, 江勇, 徐恪等. 软件定义网络系统中面向流的调度算法. 计算机学报, 2016, 39(6): 1028-1223)

[60] Li Long, Fu Bin-Zhang, Chen Ming-Yu, et al. Nimble: A fast flow scheduling strategy for OpenFlow networks. Chinese Journal of Computers, 2015, 38(5): 1056-1068(in Chinses)
(李龙,付斌章,陈明宇等. Nimble:一种适用于OpenFlow网络的快速流调度策. 计算机学报, 2015, 38(5): 1056-1068)

[61] Lu Yi-Fei, Zhu Shu-Hong, et al. Research and Implementation of TCP Congestion Control Mechanism Based on SDN in Data Center Network. Chinese Journal of Computers, 2017, 40(9): 2167-2180(in Chinses)
(陆一飞,朱书宏等. 数据中心网络下基于SDN的TCP拥塞控制机制研究与实现. 计算机学报, 2017, 40(9): 2167-2180)

[62] Xu L, Xu K, Jiang Y, Ren F, Wang H. Throughput optimization of TCP incast congestion control in large-scale datacenter networks//Computer Networks, 2017, 124(4): 46-60.

[63] Xu, L., Xu, K., Li, T., Zheng, K., Shen, M., Du, X. and Du, X. ABQ: Active Buffer Queueing in Datacenters.IEEE Network, 2020, 34(2): 232-237.

[64] Alcoz, A.G., Dietmüller, A. and Vanbever, L. SP-PIFO: Approximating Push-In First-Out Behaviors using Strict-Priority Queues//Proceedings of the Usenix Conference on Networked Systems Design and Implementation (NSDI). CA, USA, 2020: 59-76.

[65] C. Guo, H. Wu, Z. Deng, G. Soni, J. Ye, J. Padhye, and M. Lipshteyn. Rdma over commodity ethernet at scale//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Florianópolis, Brazil, 2016, pages 202–215

[66] Bosshart P, Daly D, Gibb G, Izzard M, McKeown N, Rexford J, Schlesinger C, Talayco D, Vahdat A, Varghese G, Walker D. P4: Programming protocol-independent packet processors//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Chicago, USA, 2014: 87-95

[67] Zilberman, Noa, Yury Audzevich, Georgina Kalogeridou, Neelakandan Manihatty-Bojan, Jingyun Zhang, and Andrew Moore. NetFPGA: Rapid prototyping of networking devices in open source//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). London, UK, 2015: 363-364

[68] Farrington N, Porter G, Radhakrishnan S, et al. Helios: a hybrid electrical/optical switch architecture for modular data centers//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). New Delhi, India, 2010: 339-350.

[69] Zang Da-Wei, Cao Zheng, Wang Zhan, et al. AWGR-Based OCS/EPS Hybrid Datacenter Network. Chinese Journal of Computers, 2016, 39(9):1868-1882(in Chinses)
(臧大伟, 曹政, 王展等. 基于AWGR的OCS/EPS数据中心光电混合网络. 计算机学报, 2016, 39(9):1868-1882)

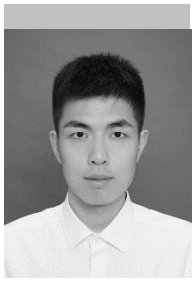
[70] Kim, D., Memaripour, A., Badam, A., Zhu, Y., Liu, H.H., Padhye, J., Raindel, S., Swanson, S., Sekar, V. and Seshan, S. Hyperloop: group-based NIC-offloading to accelerate replicated transactions in multi-tenant storage systems//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). Budapest, Hungary, 2018: 297-312

[71] Kim, D., Yu, T., Liu, H.H., Zhu, Y., Padhye, J., Raindel, S., Guo, C., Sekar, V. and Seshan, S. Freeflow: software-based virtual RDMA networking for containerized clouds//Proceedings of the Usenix Conference on Networked Systems Design and Implementation (NSDI). MA, USA, 2019: 113-125

[72] Dong M, Li Q, Zarchy D, Godfrey PB, Schapira M. PCC: Re-architecting congestion control for consistent high performance//Proceedings of the Usenix Conference on Networked Systems Design and Implementation (NSDI). CA, USA, 2015: 395-408

[73] Dong, M., Meng, T., Zarchy, D., Arslan, E., Gilad, Y., Godfrey, B. and Schapira, M. PCC Vivace: Online-learning congestion control//Proceedings of the Usenix Conference on Networked Systems Design and Implementation (NSDI). WA, USA, 2018: 343-356

[74] Mao, H., Netravali, R. and Alizadeh, M. Neural adaptive video streaming with pensieve//Proceedings of the Special Interest Group on Data Communication (SIGCOMM). CA, USA, 2017: 197-210.



DU Xin-Le, born in 1996, Ph.D. candidate. His current research interest is datacenter network and Internet architecture and protocol design.

XU Ke, born in 1974, Ph.D., professor, Ph.D. supervisor. His research interests include the next generation of Internet architecture, cyberspace security and blockchain system.

LI Tong, born in 1989, Ph.D. His current research interests include network protocol, edge computing, and IoT.

ZHENG Kai born in 1978, Ph.D. His current research interests include datacenter networking, software defined (transport layer) protocols, WAN optimizations, IoT protocols.

FU Song-Tao, born in 1982, Ph.D. candidate. His current research interest is Internet architecture and protocol design.

SHEN Meng, born in 1988, Ph.D., associate professor, Ph.D. supervisor. His research interests include network security, privacy-preserving algorithms in cloud computing and Internet protocol design.

Background

To provide users with high-quality cloud services, many large Internet companies, such as Microsoft, Google and Alibaba, have built many data centers around the world. Inside the data center, tens of thousands of servers are connected through a network of data centers with high bandwidth and low latency. Thus the increasing data-intensive traffic flow pose a great challenge to the network. It is difficult for traditional TCP to satisfy the demand of data center transmission in high throughput, low delay at the same time. A survey on traffic control for the data center network is presented in this pater. Moreover, a comparison of the proposed schemes is conducted and the future trends are discussed.

This work is supported by the Huawei Technologies Entrustment Project (YBN2018065021), the National Science Foundation of China (61825204, 61932016, 61972039), the National Key R&D Program of China (2018YFB0803405), the Beijing Outstanding Young Scientist Program (BJJWZYJH01201910003011), the Beijing Municipal Natural Science Foundation (4192050) and PCL Future Greater-Bay Area Network Facilities for Largescale Experiments and Applications (LZC0019).