网络协议和拥塞控制综述

黄修浩

摘 要 自 Google 公司提出云计算概念后，许多大型公司陆续推出了云计算服务平台，如 Google 的 GAE 平台、IBM 的“蓝云计划”、Amazon 公司的 ElasticComputingCloud 平台等。随着云计算的兴起，数据中心作为云计算载体已成为新的研究热点。数据中心由交换机和大量的服务器构成，能为信息系统中海量数据的快速存储和高效处理提供强有力的后盾。数据中心中的服务器通过网络交换机连接，形成了一个支撑数据中心内部大规模分布式服务器间流量通信的网络，称为数据中心网络(DataCenterNetworks，DCN)。传输控制协议(TCP)是目前互联网中使用最广泛的传输协议，在 DCN 数据传输中 TCP 流量约占 99%。然而，DCN 中多对一的数据传输模式会导致 TCPIncast 问题，造成网络拥塞和吞吐量崩溃，给云服务提供商带来了前所未有的压力。Incast 中的异常 TCP 行为增加了系统响应时间，对许多上层应用的服务质量产生不良影响，不可避免地降低了基于云的系统部署的适用性。传统 TCP 的设计目标已经无法满足 DCN 的结构和需求，采用更新的传输技术实现更好的拥塞控制已成为当前 DCN 急需研究的问题。

Homa/Linux 是一个实现 Homa 传输协议的 Linux 内核模块。对 Homa/Linux 的测量再次证实了 Homa 与 TCP 和 DCTCP 相比的卓越性能。在具有 40 个节点的集群基准测试中，Homa/Linux 为所有消息大小提供的延迟都低于 TCP 和 DCTCP; 对于短消息，Homa 的第 99 百分位尾部延迟比 TCP 和 DCTCP 低 7－83 倍。基准测试还表明，Homa 已经消除了网络拥塞这一重大的性能限制。尾部延迟和吞吐量现在都受到软件开销的限制，特别是由于内核之间的协议堆栈负载平衡不完善而导致的软件拥塞。如果将来可以消除软件开销，则可以实现性能提高 5－10 倍的另一个因素。

关键词：拥塞控制 网络传输协议 Homa

1 引言

网络协议指的是计算机网络中互相通信的对等实体之间交换信息时所必须遵守的规则的集合。

对等实体通常是指计算机网络体系结构中处于相同层次的信息单元。一般系统网络协议包括五个部分：通信环境，传输服务，词汇表，信息的编码格式，时序、规则和过程。1969 年美国国防部建立最早的网络——阿帕计算机网络时，发布了一组计算机通信协议的军用标准，它包括了五个协议，习惯上以其中的 TCP 和 IP 两个协议作为这组协议的通称。

TCP/IP 是因特网的正式网络协议，是一组在许多独立主机系统之间提供互联功能的协议，规范因特网上所有计算机互联时的传输、解释、执行、互操作，解决计算机系统的互联、互通、操作性，是被公认的网络通信协议的国际工业标准。TCP/IP 是分组交换协议，信息被分成多个分组在网上传输，到达接收方后再把这些分组重新组合成原来的信息。除 TCP/IP 外，常用的网络协议还有 PPP、SLIP 等。

由于网络节点之间联系的复杂性，在制定协议时，通常把复杂成分分解成一些简单成分，然后再将它们复合起来。最常用的复合技术就是层次方式，网络协议的层次结构如下：

（1）结构中的每一层都规定有明确的服务及接口标准。

（2）把用户的应用程序作为最高层

（3）除了最高层外，中间的每一层都向上一层提供服务，同时又是下一层的用户。

（4）把物理通信线路作为最低层，它使用从最高层传送来的参数，是提供服务的基础。

随着数据中心越来血多，传统的 TCP/IP 协议族成为数据中心的性能瓶颈之一，新的需求催生出新的网络协议，而 ATC21 的一篇文章要说的就是这么一种协议。

Montazeri 等人最近为数据中心引入了一种新的网络传输协议，称为 Homa。Homa 使用网络优先级队列和接收方驱动的数据包调度来支持较短的消息并消除主机下行链路的拥塞。Homa 可显著降低延迟，特别是对于高网络负载下的短消息。

Montazeri等人证明，尾部延迟比包括TCP，DCTCP，Infiniband，HULL，PIAS 和 NDP 在内的传输协议的最佳先前测量值高出近 100 倍。

然而，对 Homa 的原始评估是在研究环境中完成的，部分基于模拟，部分基于 RAMCloud 存储系统中的用户级实现（带有内核旁路）。该实现不支持通用应用程序，并且很难将其与内核 TCP 实现公平地进行比较，因为 Homa 的用户级实现避免了内核对 TCP 施加的高软件开销。因此，最初的 Homa 工作留下了悬而未决的问题，即该协议的好处是否可以在更实际的环境中实现。

ATC21 的一篇文章介绍了 Homa/Linux，一个实现 Homa 的 Linux 内核模块。我带着三个目标进行了 Homa/Linux 的开发：首先，了解内核实现的开销如何影响 Homa 的实现和性能;其次，衡量 Homa 的内核版本与成熟且广泛使用的 TCP 和 DCTCP 实现相比的性能;第三，创建 Homa 的实际实现。 鼓励在实际应用程序中使用它，并评估其对生产数据中心工作负载的好处。

该文章作了两点贡献。首先，Homa/Linux 证明了构建 Homa 的竞争性生产实现是可能的。Homa/Linux 的表现远远超过 TCP 和 DCTCP，证实了[25]中的结果。使用四个 Montazeri 工作负载，并考虑高网络负载下的短消息，Homa /Linux 下的 P99（第 99 百分位）延迟比 TCP 和 DCTCP 低 7－83 倍。与 TCP 或 DCTCP 相比，Homa 的所有消息大小都比 TCP 或 DCTCP 的延迟要低，两者的中位数均为 P99。在大多数情况下，Homa 的 P99 延迟低于 TCP 和 DCTCP 的中位数。Homa 只需要几个优先级即可实现高性能，即使只有一个优先级，它的性能也优于 TCP 和 DCTCP。

其次，这项工作提供了一个案例研究，说明在 Linux 内核的高开销环境中构建高性能传输协议所面临的挑战。Homa/Linux 必须解决各种问题，例如批处理、负载平衡和实时处理。Homa 消除了几乎所有的网络拥塞，但随着越来越多的核心必须利用来跟上不断增长的网络速度，软件拥塞正变得越来越成问题。在 Homa/Linux 中，软件开销现在是限制性能的主要因素。本文量化了这些开销。例如，在两个方向上驱动 100 Gbps 网络至少需要 18 个内核，而跨多个内核分配协议处理会将软件开销增加 2－3 倍。

尽管 Homa/Linux 提供的性能比 TCP 好得多，但其延迟和小消息吞吐量仍比原始网络速度差 5－10 倍。第 7 节认为，只要在软件中实现传输，这种差距就无法显着缩小。为了充分利用未来网络的性能潜力， 可能需要 在硬件中实现 Homa 等协议。这将需要开发新的 NIC 体系结构。

2 原理和优势

2.1Homa

Montazeri等人为数据中心引入的一种新的网络传输协议。Homa 使用网络优先级队列和接收方驱动的数据包调度来支持较短的消息并消除主机下行链路的拥塞。Homa 可显著降低延迟，特别是对于高网络负载下的短消息。Montazeri在研究环境下证明Homa尾部延迟比包括很多协议的最佳先前测量值高出近100倍，这个基于模拟和RAM-Cloud存储系统的内核旁路的用户级实现的结果留下了一些问题。不适合通用应用程序，很难与内核TCP实现公平的比较。因此最初的Homa并不确定该协议的好处是否可以在更实际的环境中实现。

2.2Homa/Linux

是一个实现了Homa传输协议的Linux内核模块。和TCP、DCTCP相比性能很好，任何规模的信息传输Homa/Linux的延迟都更低，其中短消息传输的尾延迟99分位数会比TCP、DCTCP低7-83倍。Benchmarks表明Homa消除了网络拥塞对性能的限制。不均衡的跨核协议栈带来的软件拥塞尚未解决，如果解决这个问题的软件开销会带来5-10倍的性能提升（Homa的延迟和短消息吞吐量比原始网络速度差5-10倍）

2.3Homa的关键元素

SRPT(shortest remaining processing time)：采用的是最短剩余处理时间的近似，具体策略为按长度构建优先队列，短的优先级高，且采用非抢占式。新的短消息到达了还是先处理完正在处理的长消息。

接收方驱动的包调度：最开始发送方发RTTbypes(一个RTT可以传输的字节数）大小的unscheduled包，接收方选择部分发送方进行授权，得到授权的发送方才可以发送scheduled包。

网络内优先级队列：利用现代交换机的优先队列将消息分为两组优先级（scheduled-lower和unscheduled-higher），组内按长度划分优先级，短的优先级高。

发送端SRPT：如果发送端有多个message要发，就挑选其中最短的那个，同时限制传出数据包传送到NIC

的速率避免在 发送端NIC中建立起较长的队列增加短消息的时延，与发送端配合达到近似的全局SRPT（在CPU中挑选最短的消息放入NIC的队列）。

过度提交：为了充分利用下行带宽，同时向多个peer授权，但是授权的太多又会造成交换机缓冲区积累，通常取5-10，达到利用带宽和缓冲区溢出之间平衡。这个数值是后续要提到的活动消息的个数。

2.4实现障碍

通过协议栈推送数据包的高成本:由于Linux的协议层很多，通过协议栈推送数据包的软件开销很大，批处理可以分摊开销，但是批处理会导致延迟（第一个等最后一个）

单个内核太慢:网络的速度正在迅速提高而CPU速度却没有，传输协议需要在大量内核之间平衡负载，热点形式的软件拥塞仍然是Homa/Linux尾部延迟的最大来源

实时处理:传输速率限制器需要实时处理

2.4.1 数据包流和批处理

Homa在协议栈中位于IP层上层与TCP，UDP同级。

Homa发送数据使用TSO(TCP Segmentation Offload)，模拟TSO所依赖的TCP字段，在NIC进行批处理。但是TSO只对同一个消息的数据包批处理，短消息和控制数据包必须独立的遍历协议栈

Homa接收数据时，NAPI层轮询NIC获取更多数据包，使用GRO (Generic Receive Offload)将它们组织成批处理再转发到SoftIRQ。GRO的批处理在不同消息的数据包之前也存在隔离，这会使短消息单独批处理，Homa/Linux打破了这种隔离。

2.4.2 负载均衡

数据包输出:输出堆栈完全在发送线程的核心上执行，每个核心都有一个单独的NIC通道，Linux调度程序在内核之间平衡线程，直接分配好了数据包发送负载。

数据包输入:Linux中使用两种不同的负载均衡器。运行在NIC上的RSS使用数据包标头字段的hash值在内核之间分发传入数据包，运行在NAPI层的第二个负载均衡器将NIC传入的数据包分批次后，再使用hash散列给SoftIRQ核心。这两次hash散列都会导致热点问题，Homa的策略是使用套接字流表选择最长时间未被访问的SoftIRQ内核。

负载均衡和降低低负载时的性能，Linux很难做到负载平衡配置随负载动态变化。Homa的P99延迟低负载下比高负载下更差。

2.4.3 实时处理

Pacer：限制 NIC 的内部传输队列的长度，以便在输出期间强制实施 SRPT。当 NIC 队列长度超过阈值时 ，Homa/Linux 不再立即传输数据包，消息被添加到队列中，Pacer线程被唤醒并管理队列，在 NIC 队列长度允许的情况下，按 SRPT 顺序将数据包传递到 NIC。

nic_empty_time:由于NIC队列长度无法观测，Homa维护nic_empty_time来执行SRTP。每个包被发送到IP栈上时都会根据其发送时间和包长度更新该值，作为队列空闲时间的估计。授权

2.4.4 2 级优先队列

具有非空消息的peer组成的队列，以及来自每个peer的消息队列。Peer队列中按其所对应的消息队列的剩余消息长度决定优先级，消息队列中按长短决定优先级。最高优先级的几个peer的高优先级消息处于活动状态，可以发送授权数据包，最小增量为10000（TSO批处理数据成组传输，为每个数据包授权开销大且没必要）

3 研究进展

3.1 数据中心网络Incast问题概述

TCPIncast 问题是指在 DCN 多对一的数据传输模式下造成的 TCP 传输性能下降的问题 [8]。该问题可通过如下具体的场景进行描述:客户端通过交换机与数据中心内多个服务器相连，当客户端向服务器请求某个数据块后，服务器响应该请求。客户端请求的数据块分散存储在数据中心多台存储服务器上，服务器向客户端发送的响应数据包就是这些分散的数据片，称为服务请求图 1TCPIncast 场景单元(ServerRequestUnit，SRU)。整个数据传输过程中，客户端在全部接收到请求的数据后，才会开始下一个服务请求，此过程也称为同步阻碍请求(BarrierSynchronizedRequests)。这种场景下，多个服务器向一个客户端并行传输数据，如果服务器数目过多，与客户端相连接的交换机端口的缓冲区就会溢出，造成数据包被丢弃或重传，从而出现网络拥塞和吞吐量的严重下降。

3.2 数据中心网络拥塞控制技术分类

目前研究人员已经提出了许多方案，对 DCN 中交换机、服务器等结点进行优化，以达到缓解拥塞的目的。方案可分为:基于 TCP 拥塞控制进行传输协议改进，通过流调度进行流量优化，基于 SDN 的控制方案。下面分别进行介绍和分析。

3.2.1 基于 TCP 拥塞控制的协议改进

早期的 TCP 协议及其变化版本，大多适用于用户环境比较简单、网络时延要求不高的广域网，并不适应低时延、高并发的 DCN 环境，根据 DCN 的特点，基于 TCP 拥塞控制进行协议改进，可以降低 TCPIncast 问题发生的概率，达到缓解拥塞的目的。斯坦福和微软研究院的学者发现，TCP 容易导致交换机缓存被长数据流占满，使短数据流被迫排队等待进而产生高时延，于是提出了 DCTCP 改善 TCP 在数据中心中的缺陷。

DCTCP 利用显式拥塞通告(ECN)，结合数据发送端的控制策略，以保证交换机缓存中的数据量始终低于某个阈值。其具体工作过程如下:当发现交换机队列长度超过某一阀值时，为通过队列的 TCP 数据包加上 CE 标志，告诉接收端当前网络发生了拥塞，而接收端在回复的 ACK 报文中设置 ECE 标志，通知发送方调节发送窗口。发送方的调节窗口策略为

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

其中 α 为队列拥塞程度。由于发送端会根据接收到的 ECE 标记来减小拥塞窗口，降低发送速率，所以交换机中的缓存数据也会减少，避免了短数据流在交换机中的排队，也减少了传输错误时的超时重传，因此 DCTCP 能够成功缓解网络拥塞。DCTCP 的缺点是对数据流不划分优先级，不关注流的截止时间，无法满足背景流及短流对不同截止时间的需求。

D2TCP 在 DCTCP 的基础上考虑了流的紧急程度，将流截止时间和拥塞程度一起作为发送窗口控制的依据，与 DCTCP 一样，使用 α 来测量拥塞程度，同时定义一个截止时间紧迫因子 d，计算校正值 $P(P = \alpha^d)$, P 值确定后按照以下方式调整拥塞窗口w的大小:

$$w = \begin{cases} w \times \left(1 - \frac{P}{2}\right), if\ P > 0 \\ w + 1, if\ P = 0 \end{cases}$$

由于引入了时间因子，D2TCP 能使紧急流避免错过其最后截止时间，性能上较 DCTCP 更优。

ICTCP 是一种基于主动控制思想的传输层解决方案，其拥塞避免机制部署在接收端。接收端依据其获取的可用剩余带宽来调整每个 TCP 连接的接收窗口大小，从而控制 TCP 吞吐量，防止网络拥塞。可用带宽的计算公式为

$$BW_A = max(0,\ \alpha * C - BW_t)$$

式中，α 为约束窗口参数，C 为接收端服务器接口的链路容量，BWt 为接收端接口观测到的流入总流量的带宽。ICTCP 是基于接收窗口的拥塞控制算法，接收窗口的大小设置需要权衡，窗口太小会限制 TCP 的性能，窗口太大则会降低缓解 Incast 拥塞的效果。

DCQCN 是一种适用于 RoCEv2 协议的端到端拥塞控制方案，方案包括 CP(CongestionPoint)算法、NP(NotificationPoint)算法和 RP(ReactionPoint)算法。CP 算法部署在交换机上，依据交换机队列长度判断网络拥塞状况并告知接收端;NP 算法部署在接收端，根据一定的策略生成拥塞通知数据包(CNP)，在某一时间周期内，最多只发送一个 CNP;RP 算法部署在发送端，当收到 CNP 时，参照 DCTCP 中的方式减小发送速率。DCQCN 在网卡上实现，可以保证快速公平收敛，实现高带宽利用率，实用性比较强。

Timely是一种在RDMA环境下实现的反馈式拥塞控制方案。Timely 以数据包的往返时间(RTT)作为拥塞信号，通过测量获取 RTT 然后发送给控制中心，控制中心运行核心拥塞控制算法生成更新的目标速率值，从而对数据发送速率进行控制。Timely 基于端到端的 RTT 测量，不需要从交换机那里获取队列长度等拥塞信息，可以更好地感知网络拥塞状况，实际部署的难度也相对较低。

HPCC 是阿里巴巴在高速网络拥塞控制协议研究方面的最新成果，该研究论文已被 SIGCOMM2019 收录，其核心思想是利用网络遥测技术(INT)获取精确的链路负载信息，同时计算出合适的发送速率，以实现对流量的精确控制。HPCC 以数据包的 ACK 作为速率更新的驱动，可以快速收敛，在避免拥塞的同时有效利用了空闲带宽，可以保持接近零的网络内队列，实现微秒级的超低延迟。该方案还具有公平性佳、易于在硬件上部署的优点。

3.2.2 基于流调度的控制方案

DCN 流调度的目的在于优化数据流的传输，优化目标包括减少流量完成时间(FCT)、最大程度满足具有截止时间的数据流、提高网络资源利用率等。通过合理划分数据流的优先级来分配带宽资源，流调度能有效改善 DCN 的流量传输，避免数据链路发生拥塞。

PDQ 旨在流的截止时间内尽快完成流传输，它提供了一种分布式的可执行动态调度算法，允许一组交换机协同收集流工作负载的相关信息，然后生成一个稳定的分配决策。PDQ 在交换机和收发端处都设置了控制点，发送端将发送速率、流长度、流截止时间、RTT 等流相关变量发送至交换机，接收端将收到的信息复制到 ACK 包返回给发送端，发送端根据反馈的信息来更新变量，如果流量需求超过了网络容量，发送端会停止超过截止时间的流。交换机处实施两种调度策略:如果流有截至时间，实施最早完成流优先(EDF)策略，如果流没有截止时间，则实施最短流优先(SDF)策略。PDQ 实现了全局性的平均流量传输最优，但这种流调度是抢占式的，牺牲了公平性，截止时间长的流的带宽需求不容易得到满足，而且交换机的任务复杂，导致该方案的实用难度较大。

pFabric 在发送端和交换机端同时实施拥塞控制，将流量调度与速率控制解耦。对于流调度，发送端根据流剩余大小或截止时间，在每个数据包的头部设置一个优先级编号，当数据包传输至交换机时，交换机只按照基本的优先级原则来进行流量调度。pFabric 在交换机处设置较小的缓冲区，交换机总是先传输优先级最高的数据包，当有新数据包传入且缓冲区为满时，将替换掉优先级最低的数据包。pFabric 给予短流更高的优先级，因为 DCN 中的短流更能决定用户感知和整体时延。pFabric 的速率控制的机制也相对简单，仅当数据包处于大量持续丢失状态下，才会对发送速率进行调整。其不足是需要提前获知数据流的大小或截止时间，增加了实现的难度。

有学者认为 DCTCP、PDQ、pFabric 等方案各有优势且相互间能很好地互补，于 2014 年提出一个综合了这些方案优点的网络传输控制框架 PASE。PASE 集中了自适应终端、网络内部优先级、仲裁器三种不同的传输策略，运用分布式仲裁第 6 期王远:数据中心网络拥塞控制研究综述 717 器计算流调度顺序，将剩余大小最小的流设置为高优先级，同时根据数据流的传输速率来分配带宽。在不改变网络硬件的情况下，PASE 实现了良好的网络传输性能，并且适用于各种应用工作负载和网络设置中。但是 PASE 仍需要提前获取所有流的大小和截止时间，因此实践起来有一定困难。

为了解决 pFabric 与 PASE 需要提前获取流信息的问题，另有学者于 2015 年提出了 PIAS 方案。PIAS 模拟 SJF(最短作业优先调度算法)，利用交换机中可用的多个优先级队列来实现多级反馈队列(MLFQ)，其中 PIAS 流根据当前累计发送的字节数进行优先级划分，优先级会随着累计发送字节的增加而降级。根据累计发送字节确定优先级的优势在于可以使短流在调度中优先于长流，就无需在事先获知流大小的情况下模拟最短作业优先调度(SFJ)算法，避免了 pFabric 和 PASE 的不足，同时也比较容易实现。但是，PIAS 需要获取整体流的负载情况，以确定优先级队列之间的降级阈值，其处理高动态变化流的效果不太理想。

现有的流量优化算法大都采用人工启发式的方法，存在一定时间成本和人为错误。一些方案在流量优化中引入强化学习(DRL)技术，但目前 DRL 算法的时延过大，无法处理当前 DCN 大规模的流级别流量优化，因为在 DRL 决策生成之前短流就已经结束。文献［19］提出了一个两级 DRL 系统:周边(PS)和中枢(CS)系统，模拟动物的外围和中枢神经系统来解决延展性问题。PS 位于终端，在本地收集流信息为短流做决策，以保证最小时延。CS 拥有全局流量信息，为长流做决策的同时还干预 PS 的决策。AuTo 将 DRL 算法处理与短流处理分开，避免了 DRL 决策时延对短流的影响，实现了端到端的自动流量优化系统。

3.2.3 基于 SDN 的控制方案

软件定义网络(SoftwareDefinedNetwork，SDN)通过 OpenFlow 技术将网络设备数据面与控制面分离，从而实现网络流量的灵活控制，是最有前途的网络技术之一。目前 SDN 在数据中心领域也有一定规模的实践部署，为解决 DCN 拥塞问题开辟了全新的空间。

OpenTCP 是最早的基于 SDN 技术的网络拥塞控制架构，该架构兼容 TCP 协议，使系统能根据网络状态和流量来动态调整 TCP 的规则。OpenTCP 通过一个控制器来收集网络状态信息(如拓扑和路由信息)，以及与网络流量相关的统计数据(链路利用率和流量矩阵)，然后控制器利用聚合的数据信息，根据人工定义的拥塞控制策略来调整 TCP 参数或切换适合的 TCP 版本，随后向终端主机发送周期性的拥塞控制策略的更新消息，终端主机使用拥塞控制代理(CCA)来接收消息并修改主机上的 TCP 堆栈。TCCS 方案直接从网络获取交换机的当前队列长度，当队列长度超过阈值时，拥塞信号会被发送到 SDN 控制器。控制器从全局 TCP 数据流表中选取大象

流，然后根据大象流应该有的传输速率，估算出大象流 ACK 报文的接收窗口值，并通过控制器下发修改 ACK 接收窗口的流表。交换机根据流表自动匹配大象流的 ACK 报文，修改相应的接收窗口，以降低大象流传输速率，避免网络拥塞问题。

还有学者将强化学习引入 SDN 数据中心网络的拥塞控制。基于 Sarsa 的 SDN 数据中心拥塞控制方案引入了强化学习算法，根据数据中心网络链路的负载变化，实时智能分配流的速率，使整个网络在避免拥塞的前提下，尽可能地提高链路利用率，从而实现整个网络的拥塞控制。该方案对 Sarsa 算法进行了改进，对 Q 矩阵进行训练，再根据流的请求，利用训练得到的 Q 矩阵，进行发送速率的分配控制，整个算法易实现且具有一定的控制效果。

为了克服增强学习多维感知能力弱的缺点，有学者提出了基于深度 Q 网络(DeepQNetwork，DQN)的 SDN 数据中心拥塞控制算法，其思想简要概括如下:

② 获取待分配流的链路情况、带宽占用情况和速率需求等信息;

②将上述信息输送到已训练好的神经网络中，选择具有 Q 值最优的动作 a 执行;

③ 每条流所分配的速率都将被记录，最终形成包含所有流和速率的映射算法表。

4 总结和展望

上述拥塞控制方案多采用一些拥塞信号作为拥塞控制的依据，而不同方案选定拥塞信号的方法不同，DCTCP 基于 ECN 机制，D2TCP 基于流截至时间，Timely 基于 RTT 等，拥塞信号的数量及优劣对拥塞控制方案的效果会产生重要影响。在流调度技术中，通过设置流优先级大都可以优先满足特定类型流的性能及资源需求，但实际 DCN 的应用中会产生各种不同类型的流，同时保障这些流的资源需求相当困难，一些调度算法实现了全局的流完成时间最优化，但牺牲了公平性。后续流调度技术更应该着力解决各种不同流的混合调度问题。SDN 技术的优点在于可更加快速、精确地获取全局性的网络状态信息，如交换机队列、流相关信息等，利用这些聚合信息可以更好地调整拥塞控制算法的参数，或对发送速率进行控制。未来对 DCN 拥塞控制的研究应该注重以下几个方面:

①研究更多的拥塞信号。可选择的拥塞信号越多，越有利于根据信号制定更好的拥塞控制方案。因此，对拥塞信号做进一步研究，寻找更多的拥塞信号，是未来拥塞控制研究的一个方向。

②更新网络硬件设备。与传统协议相比，许多新型传输协议改动过大，需要配套的交换机、网卡等硬件设备的支持，而新型硬件设备也是解决网络拥塞问题的一个途径，例如哥伦比亚大学的研究人员提出在光网络中使用光空间交换机(OpticalSpaceSwitches，OSS)实现一个可以缓解网络拥塞的电路交换和分组交换的组合型架构;还有 SmartNIC，通过 FPGA 协助 CPU 处理网络负载，在节省 CPU 资源的同时改善网络延迟和整体性能;再如人工智能(AI)交换机，可以动态地使用 AI 技术进行一些算法优化和参数调整。这些新型网络设备都对 DCN 拥塞控制技术的发展起到了促进作用。

③引入机器学习和人工智能技术。传统的人工启发式流调度算法无法匹配环境的突变量，具有一定局限性。为了实现动态优化资源的目标，借鉴 AuTo 方案引入人工智能和深度机器学习技术是流量优化研究的一个新思路，但是需要着重解决算法时延过大等强化学习固有的问题。另外，Q-Learning、Sarsa 及 DQN 等强化学习或深度增强学习方法目前在 SDN 数据中心网络的拥塞控制中也有一定的应用，强化学习的算法还有很多，可以考虑对其它算法进行尝试，可能会取得更好的效果。SDN 与强化学习相结合，能提高拥塞控制效率，使拥塞控制更加适应网络状态的动态变化，有望有效缓解 TCPIncast 问题，但是方案设计相对复杂，还需要更加深入的研究。

④协议测试平台的研究。目前的 DCN 拥塞控制方案中，只有 DCTCP 和 DCQCN 在实际网络中有所部署，其它所有方案尚缺乏一个定量的测量方法，这也是目前 DCN 拥塞控制研究的一个难点，未来急需一个统一的测试平台来测量各种性能指标。因此，构建统一的测试框架，以便更好地定量研究方案性能，是未来一个重要的研究目标。

参 考 文 献

[1] 王远. 数据中心网络拥塞控制研究综述[J]. 信息工程大学学报, 2019, 20(6):6.

[2] 孙三山,汪帅,樊自甫. 软件定义网络架构下基于流调度代价的数据中心网络拥塞控制路由算法[J]. 计算机应用, 2016, 36(7):5.

[3] 高秀娥, 赵鑫. 数据中心网络拥塞控制算法的研究[J]. 信息技术, 2015, 39(4):4.

[4] 黄山, 董德尊, 白巍. 无损数据中心网络拥塞控制技术[J]. 中国计

算机学会通讯, 2018, 014(010):64-68.

[5] 张子文. 无损数据中心网络拥塞控制关键技术研究[D]. 国防科学技术大学, 2013.

[6] A Linux Kernel Implementation of the Homa Transport Protocol John Ousterhout, Stanford University

[25] B. Montazeri, Y. Li, M. Alizadeh, and J. Ousterhout. Homa: A Receiver-Driven Low-Latency Transport Protocol Using NetworkPriorities.In Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, SIGCOMM '18,pages 221—-235, New York, NY, USA, 2018.Association for Computing Machinery