软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(Suppl.(2)):61−70
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

数据中心网络基于优先级拥塞控制的最后期限可感知 TCP 协议*

叶 进, 李陶深, 葛志辉

(广西大学 计算机与电子信息学院,广西 南宁 530004)

通讯作者: 叶进, E-mail: yejin@gxu.edu.cn

摘 要: 数据中心对应用服务提出了越来越严格以及多样化的要求,例如,为使用户请求得到更多的反馈应答,大多数应用对传输延迟有最后期限的限制,使得更少的最后期限错过率成为各种协议争相比较的指标,因此出现了基于 TCP 的最后期限可感知协议.但这类协议存在优先级同步问题,即相似优先级流竞争时总是相继错过最后期限.为此,提出了基于优先级控制的两种降速协议 HPD(highest priority deceleration deadline-aware data center TCP)和 P^2D(priority probability deceleration deadline-aware data center TCP),对具有侵略性的高优先级数据流进行自适应的降速处理.实验结果表明,与已有的 D^2TCP 相比,降速协议在优先级高度同步的情况下能够减少 20%的错过率.

关键词: 数据中心网络;最后期限;传输控制协议;优先级同步

中文引用格式: 叶进,李陶深,葛志辉.数据中心网络基于优先级拥塞控制的最后期限可感知 TCP 协议,2015,26(Suppl.(2)):61−70. http://www.jos.org.cn/1000-9825/15016.htm

英文引用格式: Ye J, Li TS, Ge ZH. Priority-Based congestion control scheme for deadline-aware TCP in data center networks. Ruan Jian Xue Bao/Journal of Software, 2015,26(Suppl.(2)):61−70 (in Chinese). http://www.jos.org.cn/1000-9825/15016.htm

Priority-Based Congestion Control Scheme for Deadline-Aware TCP in Data Center Networks

YE Jin, LI Tao-Shen, GE Zhi-Hui

(School of Computer and Electronic Information, Guangxi University, Nanning 530004, China)

Abstract: In modern data centers, due to the deadline-agnostic congestion control in transmission control protocol (TCP), many deadline-sensitive flows cannot finish sending before their deadline. Therefore, providing high deadline meeting ratio becomes a critical challenge in typical OLDI (online data intensive) applications of DCN. Under the partition-aggregate workflow pattern, since all the responses of single request have the same flow deadline and size, they are likely to respond nearly at the same time, which may result in that all flows with similar priority miss their deadlines. In this paper, two kinds of deadline-aware TCP, HPD and P^2D are proposed. By using the novel deceleration, HPD and P^2D alleviate the impact of priority synchronization problem. At-scale NS2 simulations show that P^2D reduces the fraction of missed deadlines by 20% compared to D^2TCP.

Key words: data center network; deadline; transport control protocol; priority synchronization

数据中心作为一种具备高可用度、高运算性能以及存储能力的现代化信息服务基础设施,承载了众多营利、非营利的组织及部门的各项服务和应用,包括网络服务、云计算、数据库和存储业务.Google 早在 2006 年就使用了 45 万台低成本服务器架构了接近 20 个数据中心;而 Microsoft 则每 14 个月将其数据中心的服务器数量翻一倍.面对如此庞大的规模,如何保障数据中心高效的存储、运算性能并降低能源消耗,成为当前亟待解决的问题[1,2].研究者们提出雪花、FiConn 等网络结构[3]、分布存储技术[4]以及虚拟化技术等[5]来降低网络开销并提高用户服务质量.

在线密集型数据业务(online data intensive application,简称 OLDI)是数据中心的典型应用,它包括网络搜索、电子商务、广告业务以及社交网络等[6].这类业务采用分散汇聚的工作模式,分散节点将用户请求分散到各

* 基金项目: 国家自然科学基金(61462007, 61363067); 广西信息科学实验中心开放基金
收稿时间: 2014-05-02; 定稿时间: 2014-08-22

© 中国科学院软件研究所 http://www.jos.org.cn

个工作节点,汇聚节点将数据汇聚并反馈给用户.其数据流主要包括:用户发送请求的查询流、携带反馈数据的时延敏感短流、持续更新工作节点内容的长背景流.由于大多数应用中用户请求需要在指定的延迟内响应,故上述前两类流的传输时间具有最后期限(deadline)的限制.早有研究表明,数据中心网络中 99.91%的流量是由 TCP 所贡献,时延敏感的短流在数量上占据数据中心网络的大部分,但是数据中心网络的大部分链路带宽却被长背景流所占用[7].因此,Deadline-Aware Data Center TCP 面临的挑战包括:短流追求更低延迟,长流需要保持较高吞吐量,链路必须具有较高的突发流容忍能力等.

有研究者提出虚拟化网络技术,采用软件定义网络(SDN)的方案,设计出虚拟交换机以实现更加灵活和多样化的数据中心网络[8],但是更多的是在协议控制技术方面的改进.传统的传输控制协议在数据中心网络中产生的问题包括:长短混合流抵达具有先入先出(first in first out,简称 FIFO)机制的缓存队列时,易产生短流的尾部拖延[6];发送节点数量的增加超过了交换机缓存处理能力而导致丢包,恢复重传致使流完成时间加剧的 Incast 问题[9];网络本身不具备最后期限的感知能力,而应用要求网络在时限内反馈应答.种种矛盾凸显了传统 TCP 协议难以满足数据中心网络日益发展的应用需求.为了弥补传统协议无法区分不同流并采取不同传输策略的不足,已经有研究者提出了一些针对分散/汇聚工作模式的大型数据中心网络传输控制协议[10].从问题解决的方案上来看,大致分为两大类.

第 1 类是以公平共享性为准则,提高网络对突发并发流的容忍能力[6,11].如 DCTCP 采用基于拥塞程度来细微调整发送速率的方法,保持了较短的缓存队列,从而有效地避免尾部拖延并提高了链路吞吐量,但忽视了长短流的不同需求[12];ICTCP 在接收端采用类似发送端的拥塞控制策略用以解决 Incast 问题,然而在工作节点数量庞大的情况下,保护缓存溢出机制会导致其性能下降[13].

第 2 类放弃公平性,优先服务短流,以缩短其传输完成时间为目的,分别从以下 3 个方面去实现:

① 在调度策略方面[14,15],PDQ 是将 EDF(earliest deadline first)和 SJF(shortest job first)的思想融入以降低平均流完成时间作为目标的调度策略.它并不感知最后期限,却可以极大地缩短数据流的完成时间.

② 拥塞控制方面[16,17],D^2TCP 是基于 DCTCP 的改进协议,它将最后期限紧迫度合理的融合进 DCTCP 的拥塞控制策略,紧迫度高的数据流获得更高的发送速率;L^2DCT 的目标是降低流的完成时间,它借用 LAS(least attained service)调度算法思想,在不能预先得知数据流大小的情况下依然能够优先完成短信息流的传输.

③ 速率控制方面[18−20],D3 作为第一个可感知紧迫流最后期限的传输控制协议,对数据流采取具有主动性以及集中性的按需分配策略.当扇入突发时,数据流的优先级通常会剧烈反转,采用 FIFO 机制的 D3 将无法掌控局面.该协议无法与 TCP 共存,且基于时钟同步的速率控制实用性不高.

我们通过深入研究基于拥塞控制的实施策略发现,现有的 Deadline-Aware Data Center TCP 存在一个现象,即当链路占用率较高时,具有相似优先级的数据流由于互相竞争会同时错过最后期限,本文称其为优先级同步问题,当相似的高优先级数据流增加时,这个现象将更加严重.为此我们提出高优先级降速最后期限可感知传输控制协议(highest priority deceleration deadline-aware data center TCP,简称 HPD)以及优先级同步概率降速最后期限可感知传输控制协议(priority probability deceleration deadline-aware data center TCP,简称 P^2D),该协议的解决方案中沿用了 DCTCP 的拥塞控制机制和类似于 D^2TCP 中的优先级设计,并在此基础上引入相同优先级的概率降速机制对资源进行重新分配,从而达到抑制少量流满足更多最后期限流的目标,最后通过实验验证了该协议的有效性以及实用性.本文第 1 节对“优先级同步”问题进行分析,第 2 节和第 3 节分别给出 HPD 以及 P^2D 的详细设计和 NS2 下的实验仿真结果.

1 问题描述

数据中心业务流具有时限特点,因此.保证用户服务质量的关键在于满足更多数据流的最后期限.但是在数据中心网络环境下,这个目标十分难以实现,原因如下:

首先,传统的拥塞控制(TCP)和流调度策略(FIFO)并未考虑数据中心业务特殊性.由于错过最后期限的数据流会被直接放弃,故传输其所耗费的链路资源被视为一种浪费,这不但间接造成链路吞吐量的降低,而且也影响

© 中国科学院软件研究所 http://www.jos.org.cn

了应用程序的性能.为此,有研究者提出交换机上的预先带宽分配机制[21,22],该机制可以精确控制分配带宽,然而需要进行硬件订制.因此,更改现有的 TCP 协议成为简单、方便、直接并且有效的方案来应对业务流的不同需求.因此提出了一系列基于传统的 TCP 改进协议[16,17],综合考量数据流的大小及紧迫程度,给予具有最后期限的数据流不同的优先等级,通过拥塞控制算法,为更为紧迫的数据流提供更高的带宽以满足其最后期限的约束.

其次,数据中心的在线密集型应用工作在分布汇聚模式下,当收到查询流时,多个工作节点同时将包含查询结果的数据短流反馈给汇聚节点.以图 1 给出的搜索应用为例,用户发送一个带有搜索关键字的查询流给分布节点,该节点将查询流分散给底层工作节点,各个工作节点的响应可能会同时返回给顶层汇聚节点,汇聚节点将包含搜索结果的短流汇聚之后反馈给用户.工作节点的数量虽多,但同时返回的应用程序业务流种类却仅有有限的几种,相似的业务流有着相近的大小与相同的最后期限.在这种模型下将存在相同优先级流的堆积现象,即相似优先级并发而造成的优先级同步问题.现有的数据中心 TCP 是否能够依然完成预设的目标?为此我们进行了一系列实验来论证.

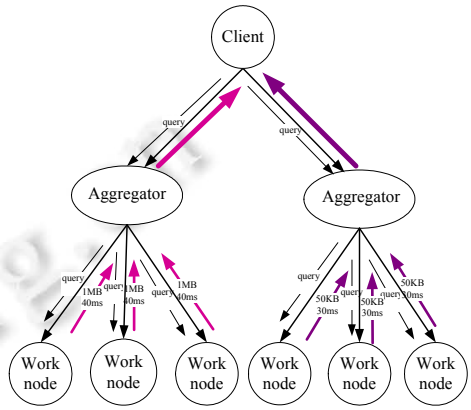
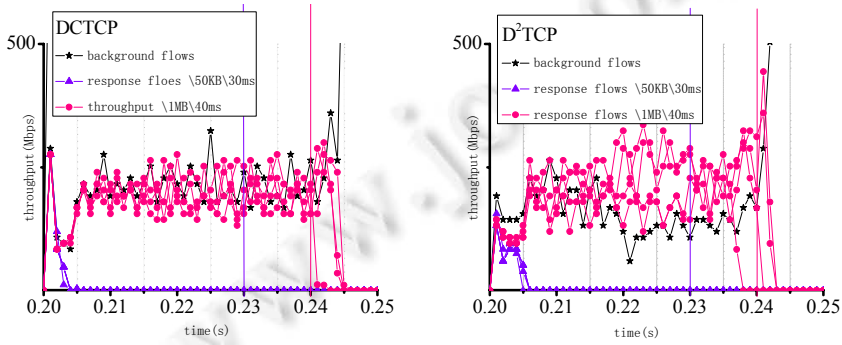


Fig.1 Web search application

图 1 Web 搜索应用

根据上述应用拓扑假设如下场景:在带宽为 1Gbps 的链路中,有大小为 10MB 周期为 80ms 的长流作为更新工作节点数据的背景流.此时,根节点的两种不同应用向工作节点发送查询流,一段时间(计算时间)后,有 9 条反馈业务流同时(0.2s)响应请求,其中 5 条大小为 50KB 最后期限为 30ms,另外 4 条大小为 1MB 最后期限为 40ms.图 2 是在 DCTCP 和 D^{2}TCP 下,不同流所获得的吞吐量对比图.DCTCP 中,1MB 的业务流全部错过了最后期限.相比之下,可感知最后期限的 D^{2}TCP 协议将大小为 1MB 的业务流作为紧迫流,并为其分配了更高的带宽.背景流吞吐量下降,50KB 流完成时间延长.但结果并未有很大的改善,依然有 3 条 1MB 的数据流错过了最后期限.通过分析我们认为这是由于相似优先级数据流在拥塞控制中发生了全局同步,相似业务流在时间上相差无几地完成传输,最终多条流错过了最后期限.



(a) 数据中心TCP (b) 最后期限可感知TCP

Fig.2 Throughput of DCTCP comparing with D^{2}TCP

图 2 DCTCP 与 D^{2}TCP 吞吐量比较

之所以会出现这种现象,本质上是由于数据中心网络瓶颈链路的资源分配供不应求,此时有些数据流因为得不到足够的资源而错过最后期限,其中就包括来自相同应用的高紧迫度数据流,它们在争夺资源由于优先级同步现象处于劣势,很难达到最后期限控制的目标.

© 中国科学院软件研究所 http://www.jos.org.cn

2 两种降速协议设计

本节提出的基于优先级的降速方案,能够估算有限的资源支持的最大最后期限敏感数据流的数量,将其作为数据流在高优先级同步情况下的资源分配策略,并设计出高优先级节点的速率调节机制,从全局分配和局部自适应调节的角度针对数据中心数据流提出一套解决方案,达到让更多的数据流在最后期限内完成传输并减少优先级同步现象的目的.本文设计的方案预期实现以下目标:(1) 在长、短流并存的情况下,链路能够短暂地为最后期限敏感的短流分配更高的带宽;(2) 当有多条高优先级流并发时,链路能够根据当前流的数量与紧迫程度来取舍高优先级的数据流,尽可能多地满足最后期限敏感流.

我们通过一个例子来说明优先级高度同步时各种不同的降速方案带来的结果.如表 1 所示,有 4 条分别来自两类应用的并发流 f_A,f_B,f_C,f_D 同时抵达瓶颈链路.我们的目的就是最小化迟到流的数量,实现最后期限的最大满足率.图 3(a)是公平共享下的传输情况.由于链路瓶颈并且最后期限紧迫,f_A,f_B,f_C,f_D 均错过了最后期限.图 3(b)所示为最后期限可感知协议下的传输,虽然 f_A,f_B 紧迫性高于 f_C,f_D 故被分配更多链路资源,f_A,f_B,f_C,f_D 依然错过了最后期限,但是从 f_A,f_B 流完成时间上看,其明显优于公平共享性的协议.图 2 所描述的正是该情形.在图 3(c)和图 3(d)中,我们假设两类应用各有一条流 f_D 和 f_B 被选取执行降速.比较两种方案,前者最终有 3 条流迟到,后者仅有 1 条流迟到,方案 2 明显优于方案 1,可见,选择哪些数据流进行降速,对缓解优先级同步所造成的最后期限错过率过高会产生不同的效果,因此,本文设计了两种基于优先级控制的概率降速方案.

Table 1 Parameters of four concurrent flows

表 1 4 条并发流相关参数

Application	Flow ID	Size	Deadline	Ts
1	f_A, f_B	1	2.5	4
2	f_C, f_D	2	5.5	6

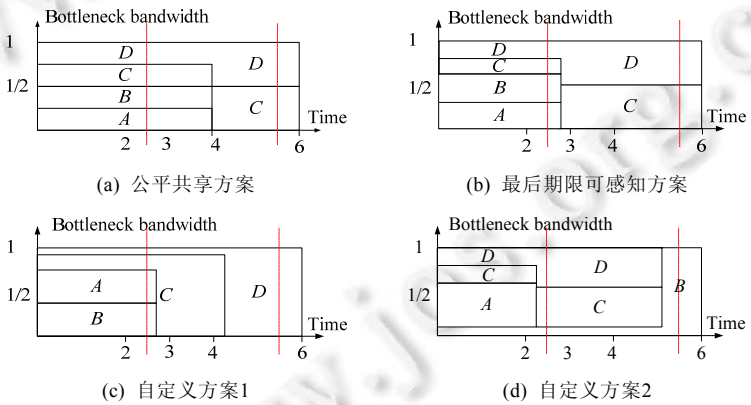


Fig.3 Resource allocation of 4 concurrent flows on the bottleneck

图 3 4 条并发流在瓶颈链路的竞争方案

2.1 HPD协议设计

HPD 方案借用汇聚处交换机的位置优势掌握全局信息,通过对所有在传数据流的资源需求进行计算分析,当发现链路资源出现供不应求的情形时,适当按照优先级从高至低的顺序选择部分数据流进行标记降速.假设数据流的大小与最后期限是可知的,第 i 条数据流要在最后期限之前完成传输其期望的窗口大小至少满足:

$$W_i' \geqslant \frac{B_i \times RTT_i}{MSS \times D_i} \tag{1}$$

其中,B_i 为数据流的剩余大小,D_i 为剩余最后期限,MSS(maximum segment size)为最大分段大小.紧迫因子 pr_i 定义为期望窗口值与常规 TCP 传输下窗口的比值,若其满拥塞状态下窗口值为 W,则平均窗口值为 $3W/4$,因此,

© 中国科学院软件研究所 http://www.jos.org.cn

$$pr_i=\frac{W_i'}{\frac{3}{4}W} \tag{2}$$

紧迫度因子 pr_i 的不同直接影响了链路的资源分配.降速概率机制就是通过对需要降速流的紧迫度因子 pr_i 进行调节,以达到抑制其获得资源,降低其发送速率的目的.已知 $S(W_1,W_2)$为窗口从 W_1 增大到 $W_2(W_2>W_1)$期间由发送方发出去的数据包数量,其间经历 W_2-W_1 个 RTT 时长.

$$S(W_1,W_2)=\frac{(W_2^2-W_1^2)}{2} \tag{3}$$

如图 4 所示,用 W_i^*表示当队列长度达到 ECN 标记阈值 K 时第 i 条流的窗口大小.在下一个 RTT 内,发送方即将收到带有 CE 码点的 ACK 包,并响应拥塞.于是拥塞窗口增加一个数据包,达到 W_i^*+1.拥塞程度函数α可以由下式确定:

$$\alpha=\frac{S(W_i^*,W_i^*+1)}{S\left((W_i^*+1)\left(1-\frac{\alpha^{pr_i}}{2}\right),W_i^*+1\right)} \tag{4}$$

将式(3)代入式(4)进行运算:

$$\alpha\approx\sqrt[pr_i+1]{\frac{2}{W_i^*}} \tag{5}$$

此时,我们可以求出振幅 A_i:

$$A_i=(W_i^*+1)-(W_i^*+1)\left(1-\frac{\alpha^{pr_i}}{2}\right)=\frac{\alpha^{pr_i}}{2}(W_i^*+1)\approx\left(\frac{2}{W_i^*}\right)^{\frac{pr_i}{pr_i+1}}(W_i^*+1)\approx 2\sqrt[pr_i+1]{\frac{2}{W_i^*}} \tag{6}$$

当网络达到稳定状态时,

$$\begin{cases}\sqrt[pr_1+1]{\frac{2}{W_1^*}}=\sqrt[pr_2+1]{\frac{2}{W_2^*}}=\ldots=\sqrt[pr_n+1]{\frac{2}{W_n^*}}=C\\ W_1^*+W_2^*+\ldots+W_n^*\leqslant W_{sum}\end{cases} \tag{7}$$

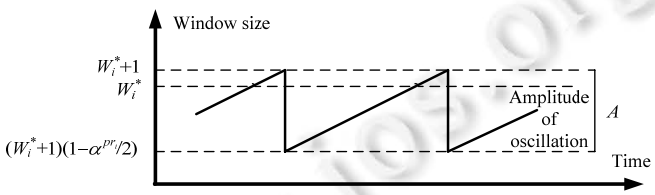


Fig.4 Varying of congestion window

图 4 拥塞窗口变化

在此基础上,可以估算出达到稳定状态时各条数据流的最大窗口预测值 W_i^*.W_{sum} 是瓶颈链路的总资源.上述资源分配策略实施步骤如下:

2.1.1 发送端

发送端保存的状态变量值包括 W_i',pr_i,以及是否决定降速的 $Flag_i$ 位,该变量初始值为 0,被通知降速的流 $Flag_i$ 位为 1.发送端将根据状态变量对数据流进行一系列预处理:若 $W'\geqslant W_{\text{sum}}$,即第 i 条数据所需资源超出链路的总资源,则发送端选择直接标记其降速;否则,当 $B_i/MSS\leqslant deadline$,即数据流即使只传送最小单位分组也依然能够在最后期限内完成数据传输时,发送端也选择直接标记其降速,这样将让出更多的链路资源给其余的流.当数据包离开时,发送端会为其附加上一个包含最新 W_i',pr_i 和 $Flag_i$ 位的调度报头.

2.1.2 交换机

在交换机上我们设计了一个针对优先级同步问题的高优先级速率调节机制.首先,HPD 交换机会将接收到

© 中国科学院软件研究所 http://www.jos.org.cn

的数据报头中流 i 所包含的状态变量保存在交换机的状态列表中,并根据式(7)计算出各条流的 W_i^*.每当有新流加入交换机时,该状态列表更新一次.HPD 交换机将会遵循从高至低的顺序选择流进行降速.W'_{sum} 为当前交换机所有流请求的总窗口值.那么,

$$W'_{\text{sum}}=\sum_{i=1}^{N}W_i' \tag{8}$$

当 $W'_{\text{sum}}>W_{\text{sum}}$ 时,表示请求的总窗口值超过了当前可提供的窗口资源总值.此时,我们将交换机内所有流的优先级由高到底进行排列,优先级最高且满足 $W_i'>W_i^*$的流 i 的 $Flag_i$ 位标记为 1.即流 i 降速,然后,

$$W'_{\text{sum}}=W'_{\text{sum}}-W_i' \tag{9}$$

此时,将状态表中流 i 的 pr_i 替换为预设的极小值,再根据式(7)计算出各条流的 W_i^*.若此时 $W'_{\text{sum}}>W_{\text{sum}}$ 依然成立,则继续选择交换机状态列表中优先级最高的流,将其 $Flag_i$ 位标记为 1,然后重复式(9),直到 $W'_{\text{sum}}\leqslant W_{\text{sum}}$.

2.1.3 接收端

接收端将复制调度报头中的信息到相应的 ACK 报文中.当接收方反馈 ACK 时,发送端最终会根据 ACK 调度报头中的信息更新该流的 $Flag_i$ 位,并确定是否需要将该流的 pr_i 位调节为一个极小值,以对其进行降速.

2.2 P²D协议描述

P²D 方案的主要思想是依据资源分配状况和数据流的优先级通过拥塞控制实现对流的自适应概率降速调节.与 HPD 精确计算选择降速流的方案不同,P²D 认为处于汇聚节处的交换机虽然可以获取大量的全局化综合信息,但是信息瞬息万变的不确定性使得交换机需要一种更具备自适应能力的选择降速方案.下面我们给出 P²D 概率选择函数的计算方式,既然交换机将全权决定如何对超出资源进行降速处理.我们定义超出资源为 W_o,期望的资源总和为 W_p,那么,

$$W_o=\sum_{i=1}^{N}W_i'-W_{\text{sum}} \tag{10}$$

假设全部数据流的集合为 $R=\{1,2,...,N\}$,R 中满足 $W_i'>W_i^*$的数据流的集合为 K,$K=\{j\in R|W_j'>W_j^*\}$,期望的资源总和为

$$W_p=\sum_{j\in K}W_j' \tag{11}$$

该类数据流的降速概率 p_i 为

$$p_i=\begin{cases}0, & W_i'\leqslant W_i^*\ ||\ W_o\leqslant 0\\ \dfrac{W_i'}{W_i^*}\times\dfrac{W_o}{W_p}, & W_i'>W_i^*且W_o>0\\ 1, & W_i'>>W_i^*且W_o>0\end{cases} \tag{12}$$

当 $W_i'>W_i^*$且 $W_o>0$ 时,将式(2)、式(7)、式(10)和式(11)带入式(12),得到:

$$p_i=\frac{3}{8}Wpr_iC^{pr_i+1}\times\frac{\sum_{i=1}^{N}pr_i-N_{\max}}{\sum_{j=1}^{M}pr_i} \tag{13}$$

$N_{\max}$ 是瓶颈链路处所能容纳的数据流的最大值.由式(13),期望值与分配值比例相差越大,该数据流被抑制的概率越高.

P²D 协议描述如下:

过程 1 是上述两种概率降速机制的伪代码.P²D 与 HPD 部署的区别主要在交换方的数据流选择方式上,我们分别使用两个降速函数来进行对比描述.

算法 1. Sender.

Initialize:

© 中国科学院软件研究所 http://www.jos.org.cn

Calculates: W_i', pr_i and $Flag_i$

if $W' \geqslant W_{\text{sum}}$, pr_i=*infinitesimal*;

else if $B_i/MSS \leqslant deadline$, pr_i=*infinitesimal*;

else if $Flag_i$=1, pr_i=*infinitesimal*;

Attach header to the packet which contain fields W_i', pr_i and $Flag_i$;

Congestion Management:

if **no congestion** *cwnd*=*cwnd*+1;

else $cwnd = cwnd \times \left(1 - \frac{\alpha^{pr_i}}{2}\right)$;

算法 2. Switch.

Receiving a packet:

Let *i* be the flow index in the list of switch;

If the flow *i* is not in the list then { //new flow happen

Update $Flag_i$, W_i', pr_i;

N=*N*+1;

Deceleration function(*N*);}

If the flow *i* is in the list then {

copy $Flag_i$ from switch to the header of packet;

update W_i' pr_i in List;}

If receiving ICMP notice from *i* then { //completing flow

remove $Flag_i$ W_i^* W_i' pr_i in switch;

N=*N*–1;

Deceleration function(*N*);}

算法 3. HPD deceleration function(*X*).

For (*i*=0; *i*<*X*; *i*++) Estimate W_i^*;

For (*i*=0; *i*<*X*; *i*++) Estimate $W'_{\text{sum}} = \sum W_i'$;

While ($W'_{\text{sum}} > W_{\text{sum}}$) {

$pr_{\max}$=0;

For (*i*=0; *i*<*X*; *i*++) {

If ($pr_i > pr_{\max}$) {

$pr_{\max} = pr_i$;

$i_{\max} = i$;} }

set $Flag_{i\max}$=1

set $pr_{i\max}$=*infinitesimal*;

$W'_{\text{sum}} = W'_{\text{sum}} - W^*_{i\max} + 1$}

For (*i*=0; *i*<*X*; *i*++)

Updates and maintains $Flag_i$, W_i^* W_i', pr_i;

算法 4. P^{2}D deceleration Function(*X*).

For (*i*=0; *i*<*X*; *i*++) Estimate W_i^*, W_o;

If W_o<0 or $W_i' \leqslant W_i^*$ maintain $flag_i$;

else if $W_i' > W_i^*$ && W_o>0 {

calculate p_i;

© 中国科学院软件研究所 http://www.jos.org.cn

set $Flag_i$=1 with p_i; }

For (i=0; i<X; i++)

Updates and maintains $Flag_i$, W_i^* W_i', pr_i;

3 性能分析与实验模拟

为了验证降速策略在处理优先级同步问题的有效性,本节采用 NS2 仿真工具进行仿真实验.模拟实验基础参数设置见表 2.长流、被降速流以及无最后期限限制的数据流的大小范围分别为 1MB~50MB,50KB~1MB,2KB~20KB,初始化其 pr_i 为 0.1.

Table 2 Simulation parameters

表 2 实验参数

Parameter	Value
Simulation time	5s
Link latency	75μs
RTO_{min}	30ms
Queue buffer	1 000pkt
RED queue size threshold	65
The latest samples weighted	1/16

3.1 同步现象应用拓扑实验

我们使用改进协议重复了本文开始部分的实验.如图 5 所示,最后期限为 30ms 的 5 条数据流以及最后期限为 40ms 的 3 条数据流均在最后期限内完成了传输,仅有 1 条最后期限为 40ms 的流被明显抑制,最终错过了最后期限.与图 2 所示的已有 Deadline-Aware Data Center TCP 相比,显著地降低了最后满足期限数据流错过率.

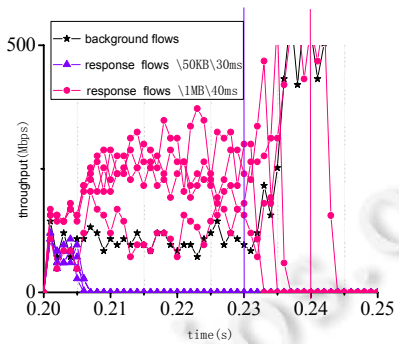


Fig.5 P^2D throughput

图 5 P^2D 吞吐量

3.2 混合交通流实验

为了确保降速协议的友好性,我们进行混合交通流的测试,实验拓扑如图 6 所示,工作节点的数量分别设定为 20,30 和 40.根节点周期性地发送查询流,当接收到查询流后,工作节点等待一个固定时间值(计算时间)后同时响应.数据流大小为 100KB~500KB,它们的最后期限为[5ms,25ms]的随机数.有两个工作节点发送不具有最后期限的长背景流,大小为 10MB 周期 80ms,查询流的数量为 1 000,其中作为对比的 D^2TCP 的 d 值被设定为 0.5~2.

图 7 和图 8 分别给出了混合交通流情况下的几种协议最后期限错过率、背景流吞吐量的对比,图 7 中传统 TCP 错过率最高,最后期限不可感知的 DCTCP 其次.而 D^2TCP,HPD 和 P^2D 表现良好,当并发流数目在 20 和 30 条时,HPD 的错过率最低,P^2D 略低于 D^2TCP.当并发流数目达到 40 时,P^2D 的错过率最低.可见,HPD 降速策略十分具有侵略性.当 N 大于 30 时,优先级同步现象明显,过激的 HPD 选择大量的高优先级流降速,其降速资源远大于实际超出资源,最终导致各条流优先级的无差别化;而 P^2D 的自适应选择方式则较为温和,P^2D 在保持较低的损失因子的同时依然能够保持较低的错过率,降低优先级同步的不良影响.同时,由图 8 可见,传统 TCP 的吞吐量

© 中国科学院软件研究所 http://www.jos.org.cn

是最低的,HPD 与 P^2D 分别低于 DCTCP 和 D^2TCP 吞吐量的 10%和 7%.这是由于 HPD,P^2D 保证短流传输策略导致的,但我们认为,以背景流吞吐量的少量牺牲换取更多高优先级数据流最后传输期限的保障是值得的.

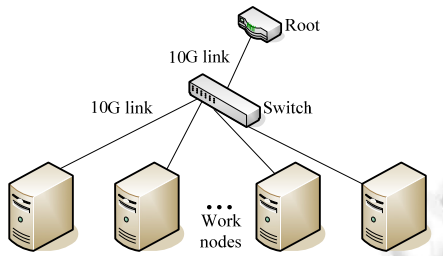


Fig.6 Network topology of mixing traffic

图 6 混合交通流实验拓扑

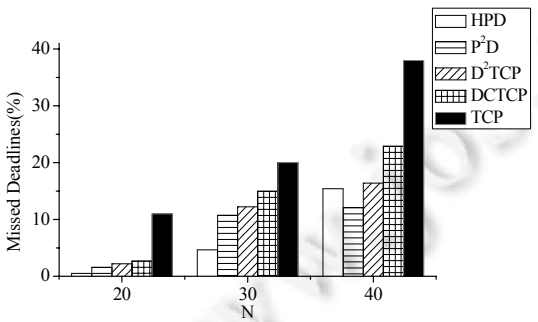


Fig.7 Ratio comparison of missed deadlines

图7 最后期限错过率对比

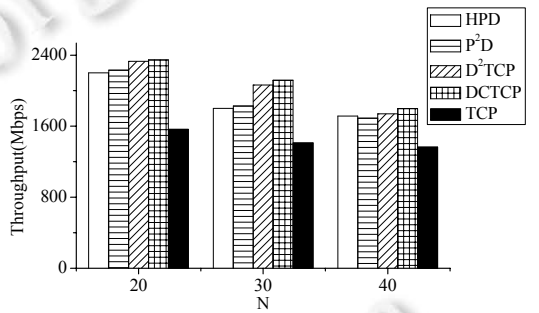


Fig.8 Throughput comparison of background flows

图8 背景流吞吐量对比

4 结 论

本文针对数据中心网络中优先级同步现象,提出了基于 TCP 协议的改进协议 HPD 与 P^2D.降速协议在链路资源有限的情形下能够对资源进行合理的分配,并对具有侵略性的高优先级数据流进行自适应的降速处理,以达到更低的最后期限错过率的目的.实验结果表明,降速协议在牺牲少量背景流吞吐量的情况下能够满足更多的短流在最后期限内完成传输,减少链路资源的浪费.下一步,我们将继续研究数据中心网络的流调度机制,以达到更精准的最后期限控制.

References:

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

[2] Li D, Chen GH, Ren FY, Jiang CL, Xu MW. Date center network research progress and trends. Chinese Journal of Computers, 2014,37(2):87−89 (in Chinese with English abstract).

[3] Liu XQ, Yang SB, Guo LM, Wang SL, Song H. Snowflake: A new-type network structure of data center. Chinese Journal of Computers, 2011,34(1):76−86 (in Chinese with English abstract).

[4] Wei LL, Chen M, Fan JH, Zhang GM, Lu ZY. Architecture of the data center network. Ruan Jian Xue Bao/Journal of Software, 2013,24(2):295−316 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4336.htm [doi: 10.3724/SP.J.1001.2013.04336]

[5] Zhang J, Ren FY, Lin C. Survey on transport control in data center networks. IEEE Network, 2013,27(4):22–26.

[6] Zheng P, Cui LZ, Wang HY, Xu M. A date placement strategy for date-intensive applications in cloud. Chinese Journal of Computers, 2010,33(8):1472−1480 (in Chinese with English abstract).

[7] Alizadeh M, Greenberg A, Maltz DA, Padhye J, Patel P, Prabhakar B, Sengupta S, Sridharan M. Data center TCP (DCTCP). In: Proc. of the SIGCOMM 2010. 2010.

© 中国科学院软件研究所 http://www.jos.org.cn

[8] Crisan D, Birke R, Cressier G, Minkenberg C, Gusat M. Got loss? Get zOVN! In: Proc. of the SIGCOMM 2013. 2013.

[9] Ren YM, Zhao Y, Liu P, Dou K, Li J. A survey on TCP Incast in data center networks. Int'l Journal of Communication Systems, 2014,27(8):1160−1172.

[10] Liu SH, Xu H, Cai ZP. Low latency datacenter networking: A short survey. arXiv preprint arXiv: 1312.3455v2, 2013.

[11] Wu HT, Feng ZQ, Guo CX, Zhang YG. ICTCP: Incast congestion control for TCP in data center networks In: Proc. of the 6th Int'l Conf. ACM, 2010.

[12] Alizadeh M, Javanmard A, Prabhakar B. Analysis of DCTCP: Stability, convergence, and fairness. In: Proc. of the ACM SIGMETRICS Joint Int'l Conf. on Measurement and Modeling of Computer Systems. ACM, 2011. 73−84.

[13] Vasudevan V, Phanishayee A, Shah H, Krevat E, Andersen DG, Ganger GR, Gibson GA, Mueller B. Safe and effective fine-grained TCP retransmissions for datacenter communication. ACM SIGCOMM Computer Communication Review, 2009,39(4): 303−314.

[14] Hong C Y, Caesar M, Godfrey P. Finishing flows quickly with preemptive scheduling. In: Proc. of the SIGCOMM 2012. 2012.

[15] Alizadeh M, Yang S, Sharif M, Katti S, McKeown N, Prabhakar B, Shenker S. pFabric: Minimal near-optimal datacenter transport. In: Proc. of the SIGCOMM 2013. 2013.

[16] Vamanan B, Hasan J, Vijaykumar TN. Deadline-Aware datacenter TCP (D^2TCP). In: Proc. of the SIGCOMM 2012. 2012.

[17] Munir A, Qazi I A, Uzmi ZA, Mushtaq A, Ismail SN, Iqbal MS, Khan B. Minimizing flow completion times in data centers. In: Proc. of the IEEE INFOCOM. 2013.

[18] Wilson C, Ballani H, Karagiannis T, Rowstron A. Better never than late: Meeting deadlines in datacenter networks. In: Proc. of the SIGCOMM 2011. 2011.

[19] Lee C, Jang K, Moon S. Reviving delay-based TCP for data centers. In: Proc. of the ACM SIGCOMM 2012 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication. ACM, 2012. 111−112.

[20] Dukkipati N, Adviser-Mckeown N. Rate control protocol (RCP): Congestion control to make flows complete quickly. Stanford University, 2008.

[21] Zhang ZW, Lü GF, Sun ZG, Wang CJ, Lu XC. Differentiated transport-layer network bandwidth allocation for InfiniBand datacenter. Chinese Journal of Computers, 2012,35(12):2505−2514 (in Chinese with English abstract).

[22] Alizadeh M, Kabbani A, Edsall T, Prabhakar B, Vahdat A, Yasuda M. Less is more: Trading a little bandwidth for ultra-low latency in the data center. In: Proc. of the NSDI. 2012.

附中文参考文献:

[2] 李丹,陈贵海,任丰原,蒋长林,徐明伟,数据中心网络的研究进展与趋势.计算机学报,2014,37(2):259−274.

[3] 刘晓茜,杨寿保,郭良敏,王淑玲,宋浒.雪花结构:一种新型数据中心网络结构计算机学报,2011,34(1):76−86.

[4] 魏祥麟,陈鸣,范建华,张国敏,卢紫毅.数据中心网络的体系结构.软件学报,2013,24(2):295−316. http://www.jos.org.cn/1000-9825/4336.htm [doi: 10.3724/SP.J.1001.2013.04336]

[6] 郑湃,崔立真,王海洋,徐猛.云计算环境下面向数据密集型应用的数据布局策略与方法.计算机学报,2010,33(8):1472−1480.

[21] 张子文,吕高峰,孙志刚,王理珺,卢锡城.面向 InfiniBand 数据中心的区别化传输层带宽划分机制.计算机学报,2012,35(12):2505−2514.

叶进(1970－),女,江苏泰兴人,博士,教授,博士生导师,主要研究领域为网络协议优化.

葛志辉(1978－),男,博士,教授,CCF 专业会员,主要研究领域为无线网络,移动计算.

李陶深(1957－),博士,教授,博士生导师,CCF 杰出会员,主要研究领域为无线网络,分布式数据库系统,云计算.

© 中国科学院软件研究所 http://www.jos.org.cn