

公平和低时延的主动队列管理算法研究

苏成龙, 金光, 江先亮, 钮俊

(宁波大学信息科学与工程学院, 浙江 宁波 315211)

摘 要: 因特网中时延敏感应用和高吞吐量应用的流行使路由器和交换机中的缓存越来越大, 导致网络流量极易出现高时延和高抖动率。基于高带宽非响应流的移动应用的普及使网络瓶颈资源利用愈加失衡, 整个网络的资源分配公平性难以得到保证。为了缓解这 2 个问题, 从时延控制和抑制高带宽非响应流抢占资源的角度, 基于 Sample-Match、L-LRU 缓存和按概率分组丢失机制, 提出了兼顾公平和低时延的主动队列管理 (FLDA) 算法。实验结果表明, 在高带宽非响应流的干扰下, FLDA 能够较好地维持队列稳定性、瓶颈链路资源利用的公平性和低时延性。

关键词: 主动队列管理; 公平性; 低时延; 队列稳定性; 高带宽非响应流

中图分类号: TP393

文献标识码: A

Active queue management algorithm focusing on fairness and low delay

SU Cheng-long, JIN Guang, JIANG Xian-liang, NIU Jun

(Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China)

Abstract: The popularity of delay sensitive applications and high throughput applications in the Internet made the caches of routers and switches larger, resulting in the high latency and high jitter rate of network traffic. With the growth of mobile applications based on the high bandwidth non-response flows, the network bottleneck resource utilization becomes more unbalanced, and the resources distribution of the whole network was difficult to be guaranteed. In order to alleviate these problems, considering the delay control and suppression of high bandwidth non-response flow, an active queue management algorithm focusing on fairness and low delay named FLDA was proposed based on Sample-Match, L-LRU cache and probabilistic packets dropping. Experimental results show that under the interference with high bandwidth non-responsive flows, FLDA can better maintain the queue stability, the fairness of resource utilization of the bottleneck link and the low delay.

Key words: active queue management, fairness, low delay, queue stability, high bandwidth non-responsive flow

1 引言

智能终端的普及和实时应用的兴起给因特网的拥塞控制带来了一系列的挑战。为了提高网络吞

吐量、避免网络分组丢失和平衡网络负载, 设备供应商趋向于在核心交换机和路由器中配备更大缓存。但是大缓存在提供数据分组缓冲功能的同时, 会造成 Buffer-Bloat^[1]问题。Buffer-Bloat 导致网络中

收稿日期: 2016-07-11; 修回日期: 2017-01-05

通信作者: 金光, jinguang@nbu.edu.cn

基金项目: 国家自然科学基金资助项目 (No.61601252); 浙江省自然科学基金资助项目 (No.LY15F020010); 软件工程国家重点实验室开放课题基金资助项目 (No.2014C31059); 浙江省科技厅公益技术应用研究基金资助项目 (No.2014C31059); 网络与交换技术国家重点实验室 (北京邮电大学) 开放课题基金资助项目 (No.SKLNST-2016-2-13)

Foundation Items: The National Natural Science Foundation of China (No.61601252), Zhejiang Provincial Natural Science Foundation of China (No.LY15F020010), The Open Project of the State Key Laboratory of Software Engineering (No.2014C31059), Zhejiang Provincial Public Technology Application Research Project (No.2014C31059), Open Foundation of State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) (No.SKLNST-2016-2-13)

很容易出现高时延和数据分组抖动等问题,甚至会降低整个网络的吞吐量。因此,当交换机或路由器配置较大缓存且网络处于高速状态,整个网络将变得极其不稳定,很难满足诸如 VoIP、在线游戏和浏览网页等交互式应用的要求。而且随着基于 UDP 协议的诸多实时应用的流行,基于 TCP 协议的响应流的性能更加无法得到保障,尤其是瓶颈链路资源的公平共享性^[2]。而公平性是多用户资源共享问题中必须加以确保的关键因素,同级用户理应获得相同份额的瓶颈链路资源。如果同级用户实际获得资源存在较大差异,部分资源保守型用户可能会由于资源的匮乏而无法完成自身任务,而资源侵占型用户可能会由于资源丰富,迅速完成任务,导致资源闲置,造成不必要的浪费。为了缓解 Buffer-Bloat 问题和瓶颈链路资源共享公平性,本文从时延控制和抑制高带宽非响应流抢占瓶颈资源等角度出发,提出了兼顾公平和低时延的主动队列管理 (FLDA, fair and low delay AQM) 算法。FLDA 在路由器/交换机的队列管理机制中设置目标排队时延和超时分组丢失机制,从而保障数据分组传输低时延,通过设置速率估计模块识别可能的高带宽非响应流并尽可能地抑制其资源侵占性,保障低带宽非响应流和响应流的瓶颈资源的正常传输,不至于因为瓶颈资源的缺乏而“饿死”。

2 相关工作

主动队列管理算法在网络传输加速上发挥着重要作用,能够有效地增强标准 TCP 的传输性能,同时能够保证流之间的公平性和可靠性。一个主动队列管理系统主要是用来控制路由器队列长度,以确保其不会处于急剧振荡状态或最大负载状态。当队列缓存处于满负荷状态,缓冲区就不能达到有效处理突发数据分组的目的,并且,此时 TCP 拥塞控制机制也不能有效地工作,除非降低发送速率。当网络流量稳定时,缓冲队列的最佳长度应该为 0,此时,任何缓冲区都可以看作是不必要的附加时延。但是面对网络突发流量时,如何缩小缓冲区的大小并且保持其队列长度的稳定性就成为了主动队列管理中的难题。为此,研究者们提出了诸多方案,这里仅对其中代表性算法进行介绍。

典型的主动队列管理算法包括 RED^[3]、BLUE^[4]、SFB^[5]、CHOKe^[6]、PIE^[7]、CoDel^[8]和 FQ-CoDel^[9]等。RED 算法利用平均队列长度和指数

加权滑动平均计算数据分组丢弃概率来维持更好的队列稳定性,但当非响应流存在时,RED 很难保证响应流的资源公平分配。为此,研究者们从抑制非响应流的角度出发,分别提出了 BLUE 和 CHOKe 算法,保证队列管理的公平性。在 BLUE 算法中,路由器视队列空闲和分组丢失事件为网络拥塞信号,通过不断更新全局标记/丢弃概率来决定数据分组的丢弃或转发。但 BLUE 引入 2 个新的参数,且参数值预先设定,缺少灵活性。之后的研究中出现了很多改进的 BLUE 算法,如 SFB 等,以完善其性能。CHOKe 算法根据一次 Sample-Match(S-M)^[6]的结果丢弃或标记数据分组,以此来惩罚高带宽非响应流,抑制其侵占性。无疑,采用多次 S-M 操作能够有效地改善 CHOKe 算法的性能,正如 gCHOKe^[10]在 CHOKe 的基础上,采用多次 S-M 机制,从而改善 CHOKe 的算法性能。然而,由于网络流量的突发性和分组丢失概率的不确定性的存在,CHOKe 算法会采取无差别分组丢失策略,极易造成响应流的数据分组的大量丢弃,对响应流的公平性和响应流的完成时间造成了不利影响,且 CHOKe 算法仅仅是依据采样匹配结果来进行后续的判断,一旦数据流量较大,采样匹配成功率降低后,极易造成缓存队列拥塞,导致排队时延剧增。

交换机的深度缓存容易导致 Buffer-Bloat 问题,增大网络传输时延。为了缓解 Buffer-Bloat 问题,时延可控的 CoDel 算法和 PIE 算法被提出,CoDel 算法以队列中数据分组的逗留时间(低于特定阈值)为依据,来决定数据分组的丢弃与否。CoDel 可以很好地维持队列的低时延性,且 CoDel 只有在逗留时间大于特定阈值时才会被执行,因此得以在诸多网络场景中应用。PIE 基于比例积分控制器严格控制数据分组在队列中的排队时延,保证时延敏感流的正常传输。它不需要对每个数据分组进行额外的处理,易于实现且不会导致高负载。另外,其参数调整是自动的,因此得以在很多网络场景中应用。另外,基于 CoDel 的 FQ-CoDel 算法采用多队列机制实现数据流的隔离,并在每个子队列上采用 CoDel 机制,以此实现更加细粒度的队列时延控制。虽然 CoDel、PIE 和 FQ-CoDel 均较好地解决了时延控制这一问题,保证了时延敏感流的及时传输,但是这三者对非响应流侵占响应流的带宽资源行为均未能加以有效控制,容易造成响应流无法获取足够的带宽资源,尤其是 TCP 长流。

综上所述, 现有算法要么侧重于抑制高带宽非响应流, 要么侧重于保证排队时延, 而本文正是在 CoDel 的基础上对采用 S-M 机制对高带宽非响应流进行区分, 实现差异化分组丢失, 保证低时延的同时兼顾公平性。而之前的工作 LRU-RC^[11]则采用 LRU(least recently used)缓存和虚拟队列在保障公平性的同时, 降低算法的复杂度, 易于实际部署。

3 FLDA 设计与实现

3.1 问题提出与解决思路

为了提高用户体验, 现有诸多网络应用不得不想方设法降低服务的响应时间。为了缩小用户发送请求和请求响应成功之间的时间跨度, 服务提供商通过在应用层利用数据压缩、内容分发等技术更快地完成业务处理。而网络提供商通过在传输层和网络层控制网络流量和规划传输路径来尽可能减小数据分组传输时间。当数据分组在路由交换网络中传输时, 节点间传输时延主要包括处理时延、排队时延和传播时延。在网络物理带宽和拓扑确定的情况下, 传播时延可以认为是确定的。而处理时延可以忽略不计, 故网络传输总时延主要取决于数据分组在路由器和交换机中停留的时间, 即排队时延。因此, 如果能够在主动队列管理算法中有效地减小排队时延, 则可以显著地减小用户请求响应时间。

而在网络公平性的研究中, 按照网络实体的级别可分为数据分组之间的公平性、流之间的公平性、链路之间的公平性、主机之间的公平性和网络之间的公平性。数据分组之间的公平性主要是指同一条流的不同数据分组之间的资源分配公平性; 流之间的公平性主要是指同一条链路上不同流之间的资源分配公平性; 链路之间的公平性主要是指同一节点对之间不同链路之间的资源分配公平性; 主机之间的公平性主要是指同一网络中主机之间的资源分配公平性; 网络之间的公平性主要是指同一网络中不同子网间的资源分配公平性。

在目前的 TCP/IP 网络和软件定义网络中, 同一条流的不同数据分组的公平性取决于该数据分组是否被丢弃, 而所有未被丢弃的数据分组之间可以看作是绝对公平的。而链路之间、主机之间和网络之间的公平性都取决于流之间的公平性, 故有效地实现流之间的公平性就能够有效地保障其他层级的公平性。

基于以上分析, 要想解决 Buffer-Bloat 造成的高

排队时延和高带宽非响应流造成的数据流间的瓶颈链路资源不公平共享问题, 必须有机地结合相应的措施来控制时延和抑制高带宽非响应流的侵占性。

本文针对 Buffer-Bloat 问题造成的高时延和高抖动率问题, 借鉴 CoDel 中的思路, 通过设置预期排队时延限制数据分组在队列中停留的时间, 从而降低数据分组的传输时延。针对公平性问题, 借鉴 CHOKe 的思路, 采用 S-M 机制识别高带宽非响应流, 并采用概率分组丢失机制来丢弃超时的高带宽非响应流的数据分组, 以此达到控制时延和保障公平性的目的。另外, 由于路由器处理每个数据分组的时间可以看作是确定的, 故当采用时延控制策略和剔除一部分容易造成排队时延不稳定的高带宽非响应流的数据分组后, 路由器中的缓存队列长度将与数据分组的排队时长成正比。因此, 如果能够维持某个路由器缓存队列长度的稳定性, 即可确保每个数据分组在该路由器中的排队时长的稳定性。而且基于数据分组排队时延的队列管理策略相对于基于队列长度的队列管理策略而言, 数据分组的超时处理能够更及时得到处理, 且使处理层次更加细粒度。每个节点上相对稳定的数据分组传输时延将会保证数据分组在整条链路上的传输时延波动较小, 从而能够缓解 Buffer-Bloat 问题带来的高抖动率问题。

3.2 测量指标

为了设计可靠高效的低时延和公平性的队列管理算法, FLDA 综合考虑时延、队列稳定性和资源公平性 3 个评价指标。队列稳定性可由队列长度的标准差表示, 定义如下

$$q_{\text{std}} = \sqrt{\frac{1}{N} \sum_{i=1}^N (q_i - q_{\text{avg}})^2} \quad (1)$$

其中, q_{std} 表示队列长度的标准差, q_i 表示瞬时队列长度, q_{avg} 表示平均队列长度。公平性可以采用 JFI(Jain's fair index)^[12]指标进行度量, 定义如下

$$JFI = \frac{(\sum_{i=1}^N x_i)^2}{N \sum_{i=1}^N x_i^2} \quad (2)$$

其中, x_i 表示流 i 的吞吐量, N 表示流的数量。JFI 取值范围为 $[\frac{1}{N}, 1]$, JFI 值越小, 公平性越低。

一个系统的绝对公平状态如式(3)所示。

$$R_i = R_j, \forall i, \forall j \quad (3)$$

其中, R_i 表示个体 i 的资源总数。AFCT(average flow complete time)^[13] 指标用来反映数据分组在网络中的停留时间, 定义如下

$$AFCT = \frac{1}{N} \sum_{i=1}^N FCT_i \quad (4)$$

$$FCT_i = T_i^a - T_i^s \quad (5)$$

其中, FCT_i 表示流 i 的完成时间, N 表示流的数量。在式(5)中, T_i^a 表示流 i 的最后一个分组的到达时间, T_i^s 表示流 i 的第一个分组的发送时间。

3.3 FLDA 主要框架

FLDA 的主要架构包括速率估计、数据分组缓存队列和队列管理等模块。速率估计模块充分借鉴了 S-M 和 L-LRU(landmark least recently used)^[14] 机制, 用于识别高带宽非响应流并缓存其相关状态信息。FLDA 的整体框架如图 1 所示。

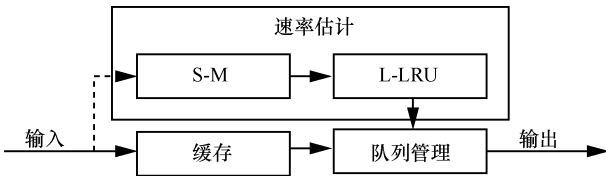


图 1 FLDA 整体框架

图 1 中, 速率估计(rate estimator)模块利用 S-M 模块和 L-LRU 缓存识别高带宽非响应流, 并记录其状态信息。当缓存在队列中的数据分组出队时, 队列管理模块通过访问 L-LRU 缓存中的记录信息丢弃或转发当前数据分组。具体处理队列管理策略如算法 1 所示。

算法 1 FLDA 算法

```

new_pkt = Pkt_arrive_Q(pkt);
flow_id = HashFunc(new_pkt);
for each i ∈ [1, N] do
    sample_packet = RandomFetchAPacket-
FromQ();
    sample_id = HashFunc(sample_packet);
    if sample_id == flow_id then
        Update_LLRLU_Cache();
    end if
end for
T = SetEnqueueTimeStame(pkt);
// when packet with flow_id dequeue
    
```

```

Dequeue(pkt);
T_out = ActualDequeueTime();
ART = T_out - T_in
drop_probability = 1 - Get_HF();
u = Random(0,1);
if u > drop_probability then
    Drop(pkt);
else
    Forward();
end if
    
```

在算法 1 中, 函数 HashFunc() 主要是对数据分组头部的五元组 (源 IP 地址、目的 IP 地址、源端口、目的端口、协议类型) 进行散列运算, 生成流标识 flow_id。相同的流经过散列运算后具有相同的流标识, 即同一条流的数据分组都具有相同的流标识。ART 表示数据分组在队列中的实际逗留时间, T_{out} 表示数据分组的实际出队时间。

当数据分组到达路由器或交换机, FLDA 首先利用函数 SetEnqueueTimeStamp() 为数据分组打上时间戳, 然后将数据分组放入缓存队列, 并将数据分组的副本送至 S-M 模块。S-M 从缓存队列中随机选取 N 个数据分组作为样本并与数据分组副本进行 N 次匹配。每条流的匹配概率如式(6)所示。

$$P_k = \frac{b_k}{\sum_{i=1}^N b_i} \quad (6)$$

其中, b_k 表示流的速率。很明显, 速率越大, 数据流被采样的概率越大, 尤其是采用多次采样^[15,16]。S-M 机制的结构如图 2 所示。

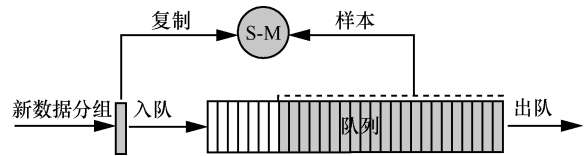


图 2 S-M 机制结构示意图

如果 N 次匹配过程中有一次待匹配数据分组和候选的样本数据分组具有相同的流标识, 则认为数据分组匹配成功, 该流就可以标识为高带宽非响应流, 可以将其状态信息存入 L-LRU 缓存。与 LRU 缓存机制不同, L-LRU 机制将缓存分为预选缓存和持续缓存。当 S-M 模块需要更新 L-LRU 缓存时, 只会更新预选缓存。而在持续缓存中, 仅将预选缓存中匹配次数大于特定阈值的相关流的状态信息写

入持续缓存，以增加流状态被挤出 L-LRU 缓存的时间。L-LRU 缓存机制的结构可参考文献[14]。

这里，采用采样匹配和 L-LRU 缓存机制识别高带宽非响应流，主要原因如下。

1) 80% 的流量产生自 20% 的高带宽非响应流^[17]，因此仅需关注后者即可，且 L-LRU 缓存的小空间就可以拥有很好的准确性。

2) 流的速率越高，被采样的概率越大，进而被 L-LRU 存储的可能性就更大。根据文献[14]的分析，只有当流的速率和持续时间均较大时才可能使其加入 L-LRU 缓存。因此，速率估计模块能够对持续时间短且带宽较高的短流和低带宽流起到一定的保护作用。具体的概率分析可参考文献[15,16]。

速率估计模块通过采样匹配和 L-LRU 缓存机制存储了最近一段时间内的高带宽非响应流的流状态信息，其中包括流的匹配次数 f_i 。依据匹配次数，可以计算流的命中频率 HF，如式(7)所示。

$$HF = \frac{f_i}{\sum_{i=1}^N f_i} \quad (7)$$

根据命中频率可计算数据分组的丢弃概率阈值，如式(8)所示。

$$drop_p = 1 - HF \quad (8)$$

随后，系统产生随机数 $\mu \in [0,1]$ ，并与 $drop_p$ 进行比较。如果 $drop_p < \mu$ ，则丢弃数据分组，反之转发。采取随机数参与分组丢失决策是为了基于随机数的随机性实现分组丢失的公平性。整个队列管理策略如图 3 所示。

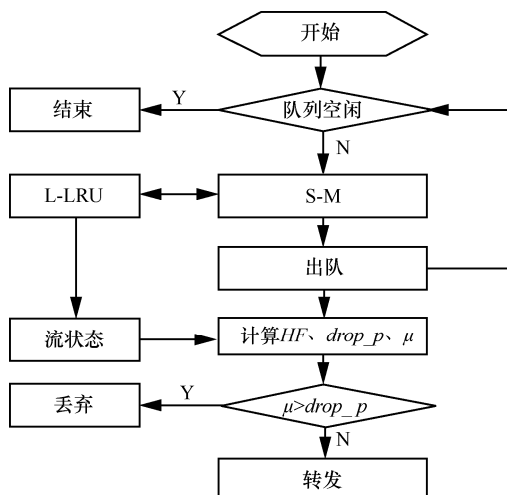


图 3 FLDA 的队列管理策略

如上所述，FLDA 致力于实现资源公平性和缓解大缓存带来的 Buffer-Bloat 问题。当高带宽非响应流占用过多带宽或队列资源，FLDA 将其数据分组大量丢弃来释放瓶颈资源。受益于时延控制机制，缓存队列长度将在预期目标附近波动，因此，数据分组的队列实际逗留时间较低，其在 AFCT 指标下也表现较好。

4 实验与分析

为了验证算法的有效性和可靠性，在高带宽非响应流的干扰下，应用 NS2 仿真实验环境，从队列稳定性、流完成时间、吞吐量和公平性 4 个角度，在被普遍认可的 Dumbbell 拓扑和 Parking-lot 拓扑中，对 FLDA、RED、CHoKe、CoDel、PIE 和 FQ-CoDel 等算法进行了一系列对比性实验。实验中，在瓶颈链路部署待测算法，在非瓶颈链路部署 DropTail 算法。S_i 和 D_i (i=1,2,3,...,n) 分别表示数据流的源节点和目的节点。

4.1 Dumbbell 拓扑实验

为了验证 FLDA 在单瓶颈链路拓扑中的性能，选取 Dumbbell 拓扑完成 FLDA 和其他算法在公平性、队列稳定性和流完成时间 3 个指标下的对比实验。Dumbbell 拓扑如图 4 所示。

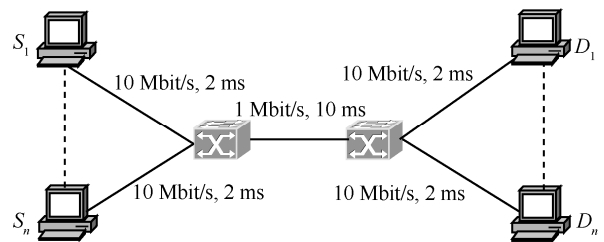


图 4 Dumbbell 单瓶颈链路实验拓扑

图 4 中设置了 2 条 UDP 流(S₁→D₁、S₂→D₂)模仿高带宽非响应流，其发送速率为 0.5 Mbit/s，数据分组大小为 500 byte。实验中所有 TCP 流开始于 0.1 s，结束于 200 s，数据分组大小为 1 000 byte 且为了制造干扰，UDP 流开始于 1.0 s，结束于 200 s，以便观察算法是否对 UDP 流的资源抢占行为有一定的抑制作用。

图 5(a)~图 5(c)是队列稳定性、公平性和吞吐量的对比实验结果。很明显，当 TCP 长流的数量从 5 增加到 50 时，就队列稳定性、公平性和吞吐量而言，FLDA 要比其他算法表现出更好的性能，其队列标准差收敛到 4，JFI 值保持在 0.9 左右，吞吐量

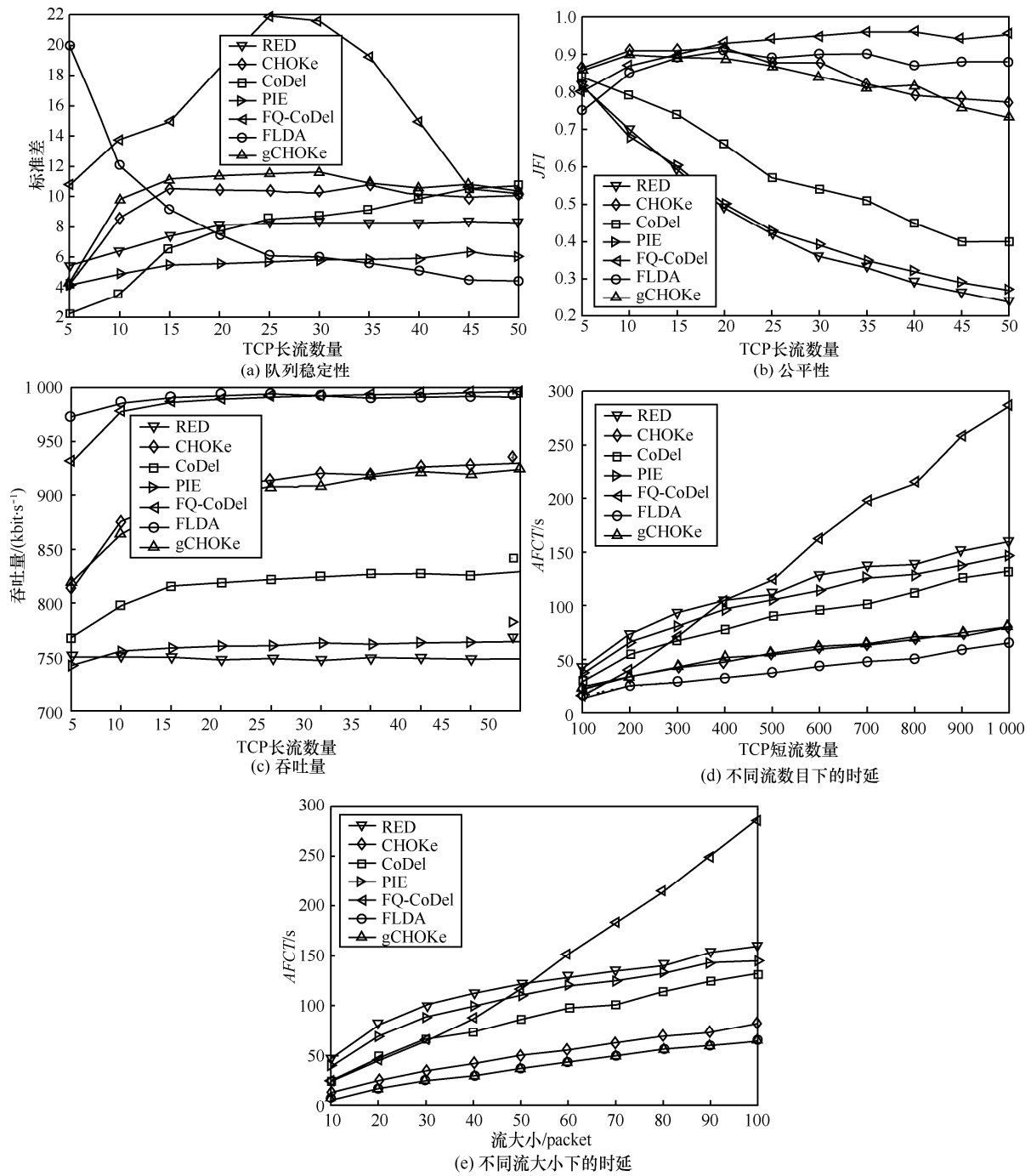


图 5 单瓶颈链路实验中对比较算法在不同指标下的性能

保持在 950 kbit/s 以上。图 5(d)和图 5(e)是 TCP 流数量变化和 TCP 流大小变化的对比性实验结果,可以看出,当 TCP 短流的数量从 100 增加到 1 000 时,FLDA 的 AFCT 值增长较其他算法慢;当 TCP 短流的大小从 10 packet 增加到 100 packet 时,其 AFCT 值增长速度依旧低于其他算法。

为了观察不同干扰流对算法性能的影响,当 UDP

流数目和大小变化时,从公平性、队列稳定性和吞吐量 3 个指标进行了验证实验,结果如图 6 所示。

从图 6 可以看出,当 UDP 发送速率由 0.1 Mbit/s 增大至 1 Mbit/s,FLDA 的 JFI 值一直保持在 0.8 以上,队列长度标准差收敛至 5 左右,吞吐量接近 1 000 kbit/s。由此可见,FLDA 在不同强度的干扰流下,依旧可以保持良好的性能。

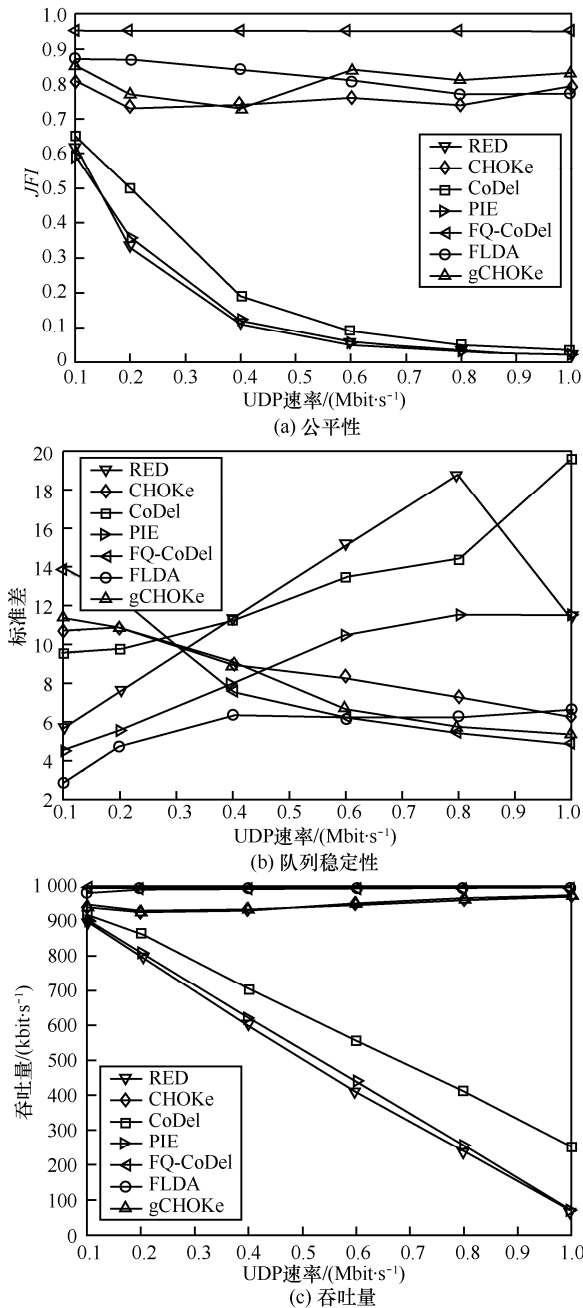


图 6 单瓶颈链路实验中对比算法在不同干扰流下的性能

4.2 Parking-lot 拓扑实验

为了验证 FLDA 在多瓶颈链路拓扑中的性能，选取 Parking-lot 拓扑完成 FLDA 和其他算法在流完成时间指标下的对比实验，以验证 FLDA 在多瓶颈链路中时延控制方面的有效性。Parking-lot 拓扑如图 7 所示。

如图 7 所示，路由器间链路($R_1 \rightarrow R_2$ 、 $R_2 \rightarrow R_3$ 、 $R_3 \rightarrow R_4$)为瓶颈链路，其队列长度为 100 packet，带宽为 10 Mbit/s，延时为 2 ms。路由器和主机间链路为非瓶颈链路，带宽为 1 Mbit/s，时延为 2 ms。所

有对比算法均部署在瓶颈链路，非瓶颈链路均部署 DropTail 算法。另外，在 C_1 、 C_2 、 C_3 和 C_4 间设置了 3 条 UDP 流($C_1 \rightarrow C_2$ 、 $C_2 \rightarrow C_3$ 、 $C_3 \rightarrow C_4$)来仿真高带宽非响应流，其发送速率为 0.5 Mbit/s，数据分组大小为 500 byte。实验结果如图 8 所示。

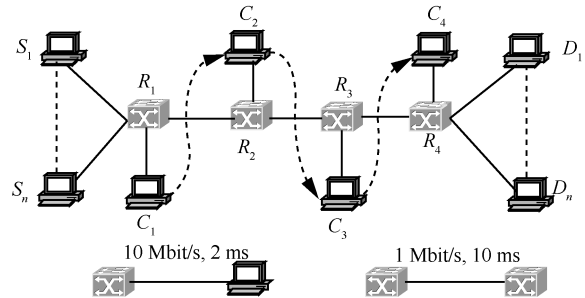


图 7 Parking-lot 多瓶颈链路实验拓扑

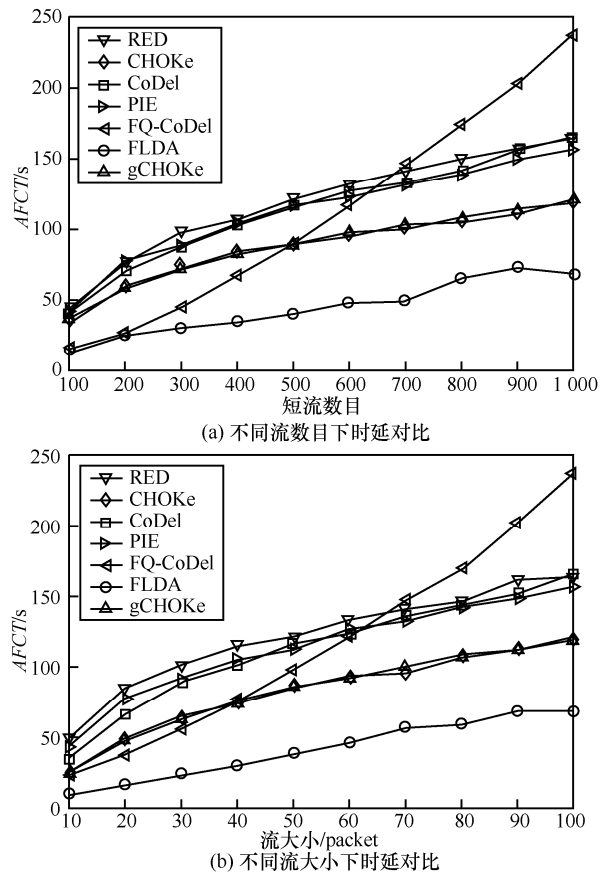


图 8 多瓶颈链路实验中的时延对比

如图 8 所示，当 TCP 短流的数量从 100 增加到 1000，FLDA 的 AFCT 值增长最慢，且其最终收敛值小于其他对比算法。这一结果说明，当 TCP 短流数量发生变化且变化较大时，FLDA 依旧可以保证数据分组具有较低的传输时延；当 TCP 流的大小由 10 个分组增加到 100 个分组，FLDA 的 AFCT 值增

长依旧最慢，其最终收敛值也小于其他对比算法。这一结果说明，当 TCP 流的大小发生变化且变化较大时，FLDA 依旧可以保证数据分组具有较低的传输时延。

另外，由于在 Parking-lot 拓扑中存在多条瓶颈链路和干扰流，每条 TCP 流经过瓶颈链路时的链路状态均不一致，所以不同算法的公平性没有可比性，且对比算法的公平性已经在单瓶颈链路中得到验证，所以此处不再进行公平性对比实验。

图 8 所示结果说明 FLDA 在多条瓶颈链路和干扰流存在的情况下，依然能够有效地减小流的完成时间，从而实现时延控制，保证算法的低时延性。

5 结束语

大缓存和高带宽非响应流的流行给网络传输和管理带来新的挑战，本文针对大缓存造成的高时延、高抖动率问题和高带宽非响应流造成的瓶颈资源公平共享问题，从时延控制和抑制高带宽非响应流抢占资源的角度出发，基于采样匹配、L-LRU 缓存和概率分组丢失机制，提出了公平和低时延的主动队列管理算法。单瓶颈链路拓扑和多瓶颈链路拓扑的实验结果表明，相较于以前的算法，在高带宽非响应流的干扰下，FLDA 能够有效地保证队列稳定性，低时延性和数据流之间瓶颈资源的公平共享，且 FLDA 实现简单，易于在实际环境中部署。在更复杂的网络场景中部署、测试和优化 FLDA 的性能将是下一步工作。

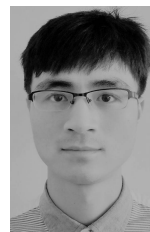
参考文献:

- [1] PAN R, NATARAJAN P, PIGLIONE C, et al. Utilizing latency control to alleviate bufferbloat: U.S. Patent 9,246,829[P]. 2016-1-26.
- [2] NOWICKI K, MALINOWSKI A, SIKORSKI M. More just measure of fairness for sharing network resources[C] //International Conference on Computer Networks. Springer International Publishing, 2016: 52-58.
- [3] FLOYD S, JACOBSON V. Random early detection gateways for congestion avoidance[J]. IEEE/ACM Transactions on Networking, 1993, 1(4): 397-413.
- [4] FENG W, SHIN K G, KANDLUR D D, et al. The blue active queue management algorithms[J]. IEEE/ACM Transactions on Networking (ToN), 2002, 10(4): 513-528.
- [5] FENG W, KANDLUR D D, SAHA D, et al. Stochastic fair blue: a queue management algorithm for enforcing fairness[C]//INFOCOM. 2001: 1520-1529.
- [6] PAN R, PRABHAKAR B, PSOUNIS K. CHOKe: a stateless active

queue management scheme for approximating fair bandwidth allocation[C]//INFOCOM. 2000: 942-951.

- [7] PAN R, NATARAJAN P, PIGLIONE C, et al. PIE: a lightweight control scheme to address the bufferbloat problem[C]// IEEE High Performance Switching and Routing (HPSR). 2013: 148-155.
- [8] NICHOLS K, JACOBSON V. Controlling queue delay[J]. Communications of the ACM, 2012, 55(7): 42-50.
- [9] HOEILAND-JOERGENSEN T, MCKENNEY P, TAHT D, et al. Flowqueue-codel[Z]. IETF Working Draft, 2014.
- [10] ESHETE A, JIANG Y. Generalizing the CHOKe flow protection[J]. Computer Networks, 2013, 57(1): 147-161.
- [11] JIANG X L, JIN G, YANG J G. LRURC: a low complexity and approximate fair active queue management algorithm for choking non-adaptive flows[J]. IEEE Communications Letters, 2015, 19(4): 545-548.
- [12] MÜHLENTHALER M, WANKA R. Fairness in academic course timetabling[J]. Annals of Operations Research, 2016, 239(1): 171-188.
- [13] CHRYSOS N, CHEN L, KACHRIS C, et al. Discharging the network from its flow control headaches: packet drops and HOL blocking[J]. IEEE/ACM Transactions on Networking (TON), 2016, 24(1): 15-28.
- [14] CHE L, QIU B, WU H R. Improvement of LRU cache for the detection and control of long-lived high bandwidth flows [J]. Computer Communications, 2005, 29(1): 103-113.
- [15] TANG A, WANG J, LOW S H. Understanding CHOKe: throughput and spatial characteristics[J]. IEEE/ACM Transactions on Networking, 2004, 12(4): 694-707.
- [16] ESHETE A, JIANG Y. On the transient behavior of CHOKe[J]. IEEE/ACM Transactions on Networking, 2014, 22(3): 875-888.
- [17] KIM J, YOON H, YEOM I. Active queue management for flow fairness and stable queue length[J]. IEEE Transactions on Parallel and Distributed Systems, 2011, 22(4): 571-579.

作者简介:



苏成龙 (1991-), 男, 安徽安庆人, 宁波大学硕士生, 主要研究方向为网络拥塞控制、无线网络和软件定义网络。

金光 (1972-), 男, 浙江台州人, 博士, 宁波大学教授、硕士生导师, 主要研究方向为网络协议、无线网络和物联网。

江先亮 (1986-), 男, 安徽黄山人, 博士, 宁波大学讲师, 主要研究方向为数据驱动传输协议、数据中心网络传输控制和无线网络协议。

钮俊 (1976-), 男, 四川阆中人, 博士, 宁波大学副教授、硕士生导师, 主要研究方向为物联网和形式化方法。