文章编号:1001-2486(2004)03-0029-05

DSOCC:面向区分服务的拥塞控制*

张明杰,朱培栋,卢锡城
(国防科技大学计算机学院,湖南 长沙 410073)

摘 要:在区分服务(DiffServ)中使用确保转发(Assured Forwarding)进行TCP传输,存在带宽分配不公平问题。已有的工作没有综合考虑订购带宽、报文大小、RTT(Round Trip Time)对TCP性能的影响。为了解决TCP在DiffServ中的公平性问题,提出了面向区分服务的拥塞控制算法DSOCC(DiffServ-Oriented Congestion Control)。通过模拟验证了该算法的有效性。

关键词:区分服务;确保转发;传输控制协议;带宽分配;公平性
中图分类号:TP393 **文献标识码**:A

DSOCC: DiffServ-Oriented Congestion Control

ZHANG Ming-jie, ZHU Pei-dong, LU Xi-cheng
(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: When using DiffServ AF to transfer TCP traffic, the bandwidth is not allocated in a fair way the previous work does not consider the three factors affecting TCP's throughput all together: subscribed bandwidth, packet size and RTT. The paper proposes DiffServ-Oriented Congestion Control (DSOCC) to resolve the problem on TCP fairness. The validity of the algorithm is verified using ns-2 simulations.

Key words: DiffServ; AF; TCP; bandwidth allocation; fairness

IETF提出的DiffServ体系结构[1]把应用需求分成几类,核心路由器只需要支持几类简单的调度转发策略,从而解决了核心路由器支持服务质量(QoS)所面对的可扩展性问题。

在DiffServ中,外部可观察的路由器对每一类报文的转发行为称为Per-Hop-Behavior(PHB)。IETF区分服务工作组定义了两类PHB:EF PHB[2]和AF PHB[3]。EF PHB主要用于实时应用;AF PHB用于带宽保证的应用。本文主要针对AF PHB。

使用AF PHB进行传输的应用首先向网络服务供应商订购一定的带宽,然后才能使用该服务。网络的边界路由器负责为用户报文打上IN或OUT标志,如果当前用户的报文流量在订购的速率CIR(Committed Information Rate)范围之内,那么就为报文打上IN标志,否则就打上OUT标志。网络中的核心路由器使用RIO(RED with IN & OUT)算法[4]进行缓冲管理。RIO算法的特点是:在发生拥塞时,首先丢弃OUT报文,然后才丢弃IN报文。

大量研究表明,TCP流在DiffServ网络中进行传输时,存在带宽分配不公平的问题[5,6]。体现在当网络资源充足时,剩余带宽不能按订购比例进行分配;当资源不足时,带宽资源不能按订购比例降级。针对该问题,人们进行了许多研究工作,主要体现在两个方面:

一方面,改进边界路由器的标记算法。已有的标记算法包括:双速率三色标记(TRTCM)[7]、时间滑动窗口三色标记(TSWTCM)[8]。这些算法没有考虑TCP的公平性问题,具有不同订购带宽、报文大小和RTT的应用获得的带宽不能按比例分配。基于公平性考虑,文献[9]提出了智能流量监管器ITC,ITC考虑了RTT和订购带宽对TCP吞吐量的影响,该方法的不足在于需要所有的边界路由器进行通信,交换RTT和订购带宽信息,实现复杂。基于TCP吞吐率公式的标记算法EBM[10]优于上面的算法,但是该算

* 收稿日期:2003-11-27
基金项目:国家自然科学基金资助项目(90204005);国家863计划资助项目(2003AA121510)
作者简介:张明杰(1974—),男,博士生。

法中，两个重要的参数 *YScale* 和 *RScale* 不能随资源变化进行动态调整，不适合动态变化的网络环境。除了改进标记算法之外，另一个努力方向是改善 TCP 本身的拥塞控制机制。本文认为，造成带宽分配不公平的直接原因来自于 TCP 的拥塞控制算法，因此对 TCP 的拥塞控制算法进行改进是最有效的方法。

1 已有工作对 TCP 性能的改进

TCP 使用加式递增积式递减算法 AIMD[11] 进行拥塞控制。经典的 TCP 算法存在不公平性：因为在拥塞避免阶段，每隔一个 RTT，窗口增加一个报文，这样使得 RTT 小的链接速率增加快；而且，同是增加一个报文，报文大的链接速率增加快。针对 TCP 在 DiffServ 中的性能，已有的改进包括：

文献[12]基于经典的 TCP 算法，对 TCP 的改进如下：每当收到 ACK 报文时，拥塞窗口 *cwnd* 增加 $c \times rtt^2/cwnd$，c 是常数。当检测到报文丢失时，慢启动阈值并不设置为 *cwnd* 的一半，而是设置为目标窗口 *RWnd*，目标窗口定义为 $RWnd = CIR \times rtt/pktSize$。当丢失的是 OUT 报文时，*cwnd* 减半；当丢失的是 IN 报文时，*cwnd* 设为 1。该算法在窗口打开时考虑了 RTT 对 TCP 性能的影响，因而当资源充足时，在订购带宽和报文大小相同的情况下能够消除 RTT 的影响；在资源不足时，由于慢启动阈值设置偏大，导致 TCP 操作在慢启动阶段性能下降，而且该算法没有考虑订购带宽和报文大小对 TCP 性能的影响。

文献[13]提出了自适应标记算法 APM，该算法中 TCP 维护两个拥塞窗口：*pwnd*（priority window）和 *bwnd*（best-effort window）。*pwnd* 记录了 IN 窗口大小，*bwnd* 记录了 OUT 窗口大小。当收到 ACK 时，如果当前流量小于订购带宽，*pwnd* 和 *bwnd* 都增加；否则 *pwnd* 减小，同时 *bwnd* 增加，即减小标记为 IN 的报文数量。当 IN 报文丢失时，说明网络中存在严重的拥塞，因此 *pwnd* 和 *bwnd* 都减半；当 OUT 报文丢失时，*bwnd* 减半，*pwnd* 保持不变。该算法在资源充足时能保证流要求的速率，但是没有考虑订购带宽，从而造成不公平。APM 算法也没有考虑 RTT 和报文大小对带宽的影响。

对 TCP 机制进行改进，标记算法应该在端系统执行，这样端才能知道丢失的报文是 IN 报文还是 OUT 报文；如果在边界路由器执行标记功能，需要把标记结果通知端系统。

2 面向区分服务的拥塞控制算法 DSOCC

针对在 DiffServ 网络中进行 TCP 传输的不公平性，本文提出了面向区分服务的拥塞控制算法 DSOCC（DiffServ-Oriented Congestion Control）。下面首先给出公平性定义。

假设瓶颈链路的带宽为 C，有 N 条 TCP 流竞争带宽，流 i 订购的带宽为 R_i。

定义 1 成比例分配剩余带宽 如果 $\sum_{i=1}^{N} R_i < C$，那么流 i 获得的最终带宽为

$$R_i + \frac{R_i}{\sum_{i=1}^{n} R_i}\left(c - \sum_{i=1}^{N} R_i\right)$$

定义 2 成比例降级 如果 $\sum_{i=1}^{N} R_i > C$，那么流 i 获得的最终带宽为

$$R_i - \frac{R_i}{\sum_{i=1}^{n} R_i}\left(\sum_{i=1}^{N} R_i - C\right)$$

2.1 DSOCC 算法

本文提出的 DSOCC 拥塞控制机制在窗口操作中综合考虑了订购带宽、报文大小以及 RTT 对 TCP 吞吐率的影响。DSOCC 算法由两部分组成：窗口打开算法和窗口关闭算法。

窗口打开算法如图 1。其中，$R = \frac{c \times \tau}{k}$，$c$ 为订购带宽，τ 为 RTT，k 为报文大小。窗口打开算法综合考虑了 CIR、RTT 和 *pktSize* 因素，主要基于下面的考虑：订购带宽越大，窗口打开速度越快；RTT 越大，窗口打开速度越快；报文越大，窗口打开速度越慢。

```
每收到一个 ACK:
if(cwnd < ssthresh){cwnd + = 1;}
else{cwnd + = α × R²/cwnd;}
```

图 1 DSOCC 拥塞窗口打开算法

Fig.1 DSOCC congestion window opening

窗口关闭算法如图 2。其中,*avgRate* 是平均速率,采用时间滑动窗口测量[4]。窗口关闭算法考虑了资源充足和资源不足两种情况,如果资源充足,IN 报文丢失的概率很小,端看到的大多数丢失为 OUT 报文;如果资源不足,*avgRate* 小于 CIR,由 RIO 算法可知,OUT 报文几乎都被丢失,IN 报文丢失的概率由订购级别决定。类似于 APM,当 OUT 报文丢失时,*cwnd* 只把与 OUT 对应的窗口部分减半;如果 IN 报文丢失,整个拥塞窗口减半。

```
检测到报文丢失:
if(avgRate > CIR){
    if(丢失的是 IN 报文){cwnd/2 = 2.0;}
    else{cwnd - = (cwnd - R)/2.0;}}
else{cwnd/ = 2.0;}
```

图 2 DSOCC 拥塞窗口关闭算法

Fig.2 DSOCC congestion window closing

2.2 DSOCC 参数选择

当 CIR = A, RTT = B, k = X 时,在一个 RTT 内拥塞窗口增长一个报文,即 $\alpha\left(\frac{A\cdot B}{X}\right)^2 = 1$,则 $\alpha = \left(\frac{X}{A\cdot B}\right)^2$。

3 模拟验证

本文使用 ns-2[14]模拟器对 DSOCC 算法进行了验证。模拟拓扑如图 3。实验中,发送源分成 5 组,每一组有 5 个相同的 FTP/TCP 源。拓扑中,每条链路的带宽为 10Mbps,除了发送源到边界路由器的延迟外,其它链路的延迟为 5ms。边界与核心路由器的 RIO 设置如表 1 所示。

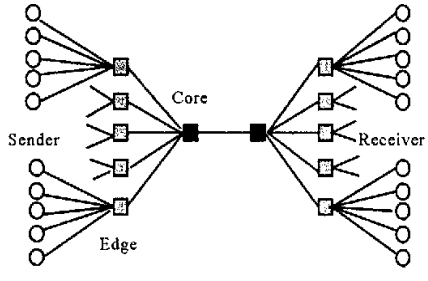

图 3 模拟拓扑

Fig.3 Simulation topology

表1 RIO参数设置

Tab.1 RIO Parameter Settings

	IN-profile	OUT-profile
$\min_{th}$	40	20
$\max_{th}$	80	40
$\max_{p}$	0.02	0.2
W_q	0.002	0.002

在图4中,算法1是指传统的TCP+TSW标记;算法2指文献[12]中所提算法;算法3指文献[13]中所提算法;算法4指DSOCC算法。每组实验进行两次,分别对应订购级别50%(代表资源充足情况)和125%(代表资源不足情况)。

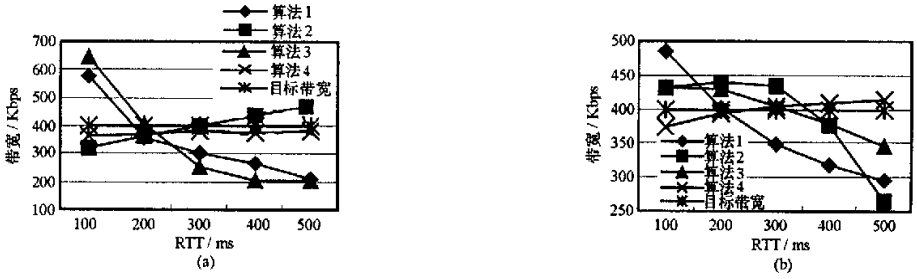

图4 (a)低定购级别(50%);(b)高订购级别(125%)

Fig.4 (a)Under-subscription (50%);(b)Over-subscription (125%)

3.1 消除RTT影响

实验中第一组源节点到边界路由器的链路延迟为50ms,第二组为100ms,第三组为150ms,第四组为200ms,第五组为250ms。每一组中5个源的平均速率作为该组的速率。RTT与所获得带宽的关系如图4(a)、(b)所示。图中,目标带宽是指流应获得的带宽,与定义1和定义2一致。从图4(a)可以看出,在资源充足时,算法2和算法4都具有比较好的公平特性;算法2与算法4的区别在于:当有OUT报文丢失时,算法2设置 *cwnd* 为 *RWnd*,而算法4设置 *cwnd* 为 $RWnd+(cwnd-RWnd)/2$,从而造成算法2性能稍差于算法4。在资源不足时,算法2性能比较差,原因在第1节中进行了分析。在资源充足时,算法3的公平性差;当资源不足时,算法3的性能稍差于算法4。

3.2 消除订购带宽的影响

第一次实验中每组节点订购的速率分别为66Kbps、132Kbps、198Kbps、264Kbps、330Kbps;第二次实验中每组节点订购的速率分别为100Kbps、300Kbps、500Kbps、700Kbps、900Kbps。订购带宽与所获得带宽的关系如图5(a)、(b)所示。从图5(a)、(b)可以看到,在资源充足时APM算法没有考虑订购带宽。在资源

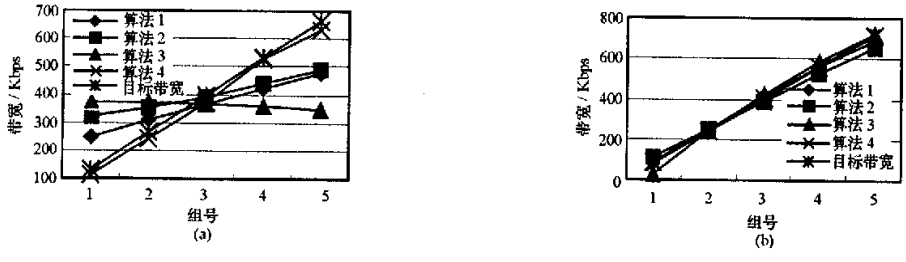

图5 (a)低定购级别(50%);(b)高订购级别(125%)

Fig.5 (a)Under-subscription (50%);(b)Over-subscription (125%)

不足时算法 1、2、3 的公平性在提高,即在资源不足时都能大致按比例下降,这一点与文献[5]的结果是一致的,但 DSOCC 算法在资源充足和资源不足的情况下都有较好的性能表现。

3.3 消除报文大小的影响

每一组源发送的报文大小分别为 300 字节、600 字节、900 字节、1200 字节、1500 字节。

报文大小与所获得带宽的关系如图 6(a)、(b)所示。从图 6(a)、(b)可以看出算法 2 在资源充足时和算法 4 相差较大,在资源不足时接近算法 4,算法 1 和算法 3 在资源充足和资源不足时都不如算法 4。这一节通过模拟验证了 DSOCC 算法的有效性,从模拟的结果看,DSOCC 算法可以达到公平带宽分配的目标。

4 结束语

本文针对 TCP 在 DiffServ 中的带宽公平分配问题提出了 DSOCC 算法,主要对 TCP 的速率增加量进行改进,在 TCP 打开拥塞窗口时综合考虑了订购带宽、报文大小和 RTT 对 TCP 吞吐量的影响。通过模拟验证了该算法的有效性,能够达到公平分配带宽的目标。对 DSOCC 进行理论分析是下一步的工作。

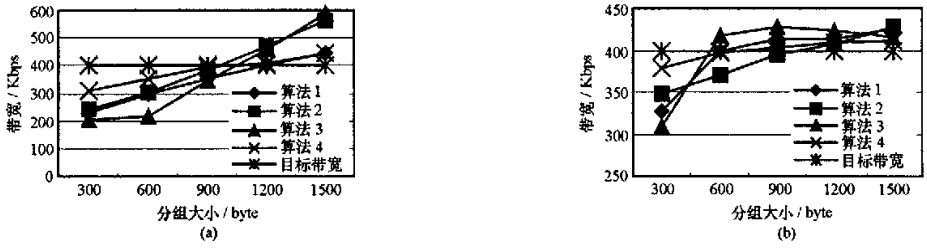

图 6 (a)低定购级别(50%);(b)高订购级别(125%)

Fig.6 (a)Under-subscription (50%);(b)Over-subscription (125%)

参考文献:

[1] Blake S, Black D, Carlson M, Davies E, Wang Z, Weiss W. An Architecture for Differentiated Services[S]. RFC 2475, December 1998.

[2] Jacobson V, Nichols K, Poduri K. An Expedited Forwarding PHB[S]. RFC 2598, June 1999.

[3] Heinanen J, Baker F, Weiss W, Wroclawski J. Assured Forwarding PHB Group[S]. RFC 2597, June 1999.

[4] Clark D, Fang W. Explicit Allocation of Best Effort Packet Delivery Service[J]. IEEE/ACM Transactions on Networking, 1998,6(4): 362 - 373.

[5] Seddigh N, Nandy B, Pieda P. Bandwidth Assurance Issues for TCP Flows in a Differentiated Services Network[C]. In Proc. of IEEE GLOBECOM'99, March 1999.

[6] Yeom I, Reddy A L N. Modeling TCP Behavior in a Differentiated Services Network[J]. IEEE/ACM Transactions on Networking, 2001,9(1):31 - 46.

[7] Heinanen J, Guerin R. A Two Rate Three Color Marker[S]. RFC 2698, September 1999.

[8] Fang W, Seddigh N, Nandy B. A Time Sliding Window Three Color Marker (TSWTCM)[S]. RFC 2859, June 2000.

[9] Nandy B, Seddigh N, Pieda P, Ethridge J. Intelligent Traffic Conditioners for Assured Forwarding Based Differentiated Services Networks[C]. In Proc. of HPN'00, May 2000.

[10] El-Gendy M A, Shin K G. Equation-based Packet Marking for Assured Forwarding Services[C]. In Proc. of IEEE INFOCOM'02, June 2002.

[11] Jacobson V. Congestion Avoidance and Control[C]. In Proc. ACM SIGCOMM'88, August 1988.

[12] Fang W, Peterson L. TCP Mechanisms for DiffServ Architecture[R]. Princeton University Technical Report, TR-605-99, September 1999.

[13] Feng W, Kandlur D, Saha D, Shin K. Adaptive Packet Marking for Maintaining End-to-end Throughput in a Differentiated-services Internet[J]. IEEE/ACM Transactions on Networking, 1999, 7(5): 685 - 697.

[14] Ns-2 Network Simulator[R]. http://www.isi.edu/nsnam/ns.