一种支持优先级标记处理的主动队列管理机制*

李方敏[1], 叶澄清[2]

[1](湘潭工学院 计算机科学系,湖南 湘潭 411201);
[2](浙江大学 计算机科学系,浙江 杭州 310027)
E-mail: lifangmin@163.net
http://www.xtpu.edu.cn

摘要: 随着 Internet 流量的日益增加,依赖平均队列长度管理拥塞控制的 RED(random early detection)队列管理算法有其内在的缺点,即使结合 IETF(Internet engineering task force)明确的拥塞通知 ECN(explicit congestion notification)也不能有效地阻止包丢失.在分析比较 RED 算法和 BLUE 算法的基础上,提出了一种加强的主动队列管理机制——EBLUE(enhanced BLUE),然后结合 EBLUE 研究了 TCP 的拥塞控制机制,并且利用令牌桶在网络入口点对注入网络的流量进行标记,以提供最小带宽保证.最后在 NS 网络仿真环境下对 EBLUE 的丢包率、链路带宽使用效率等方面进行了仿真实验和性能评价.实验结果证明所提出的机制可以有效地支持带宽预留服务.

关 键 词: 主动队列管理;拥塞控制;令牌桶标记
中图法分类号: TP393 文献标识码: A

Internet 发展到今天,取得的成功主要归功于其协议的不断改进和加强,其中 TCP/IP 协议扮演了一个非常重要的角色,它已成为 Internet 协议的事实标准.随着 WWW 应用和各种实时多媒体信息在 Internet 网上的不断增加,TCP/IP 协议的弱点也不断显现出来,因此对现有协议进行改进,使其提供服务质量支持成了网络界的一个研究热点问题,而其中一个非常重要的方面就是通过一种有效的队列管理机制提供各种 QoS.

有效的拥塞控制机制和队列管理可以使网络处于良好的运行状态,而大量的没有拥塞控制机制的流量将会导致网络崩溃.TCP 基于加法增加、乘法减少的慢速启动、拥塞避免、快速重传、快速恢复机制加上简单的 FIFO 队列管理算法,使得 Internet 在过去能健康地运行,但随着 Internet 上各种流量的不断增加,网络带宽的不足使目前的尽力传送流的丢包率大量增加,因此尽力传送流的有效性日益恶化.为此,IETF 提出使用明确的拥塞通知机制(ECN)[1,2]和主动队列管理[3]阻止包丢失,其思想是路由器在队列溢出之前通过主动队列管理机制用 ECN 通知源,而源相应地调整发送速率,这样可以避免通过包丢失来检测网络拥塞,从而将拥塞通知从包丢失中分离出来.

除了基于尽力传送的应用之外,很多应用,如基于 WWW 的多媒体应用以及基于 UDP 和 RTP 的连续实时多媒体应用一般都需要一定的带宽保证,以提供更好的服务质量,IETF 提出了基于 RSVP(resource reservation setup protocol)和相应服务类型的集成服务框架 INTSERV(integrated services),但 INTSERV 需要对现有的网络基础设施进行重大的修改,要求所有的路由器支持 RSVP 协议,并且大大加重了路由器的处理负担,为此,IETF 又提出粗粒度的区分服务框架 DIFFSERV(differentiated services),将大部分处理负荷推到边界路由器,而中间路由器只根据预先定义的每站行为(per hop behavior)进行简单的服务区分处理.

* 收稿日期: 2000-02-17; 修改日期: 2000-10-16
基金项目: 国家自然科学基金资助项目(69974031)
作者简介: 李方敏(1968－),男,湖南涟源人,博士,副教授,主要研究领域为网络服务质量,新型网络协议,多媒体通信技术; 叶澄清(1939－),男,江苏泰县人,教授,博士生导师,主要研究领域为高性能体系结构,计算机网络.

本文在分析 BLUE[4]算法的基础上提出了 EBLUE 主动队列管理机制,然后基于 IETF 定义的框架在端主机实现基于令牌桶的自适应的支持标记的速率调节,研究 TCP 支持最小速率保证的拥塞控制机制,从而支持 controlled-load 服务[5].

1 EBLUE 算法

1.1 BLUE算法

在文献[4]的第 3 节,作者提出了一种自适应的 RED 机制,该机制根据连接数目、流量的多少,在接收包时更新平均队列长度,然后根据平均队列长度动态调整 $\max_p$,这种自配置的 RED 机制能动态适应网络负载的改变,但该机制即使结合 ECN 在重负载情况下仍然不能有效地消除包丢失,只是相对原始的 RED 机制有所改善.为了消除包丢失,作者进一步改进了 TCP 协议的拥塞控制机制.

所有根据队列长度估计网络拥塞状态的队列管理机制都不同程度地存在上述问题,这是因为队列长度并不能完全反映网络的拥塞状况.即使在只有一个连接的情况下,如果源以高于网络瓶颈链路的速率发送数据就会导致拥塞.而 RED 根据平均队列长度决定网络的拥塞程度,为了适应不同的拥塞情况,RED 需要大量的参数,并合理地配置和足够的缓冲空间,只有这样 RED 才能达到一个理想的操作点.

在研究 RED 队列调度的基础上,文献[4]的第 4 节提出了一种不同于 RED 的主动队列管理机制——BLUE,它通过包丢失和链路使用历史管理拥塞,其算法如下:

算法 **1**. BLUE 算法.

(1) 包丢失事件:

```
if ((now–last_update)>freeze_timed){
    pm=pm+deltad;
    last_update=now;
}
```

(2) 链路空闲事件:

```
if ((now–last_update)>freeze_timei){
    pm=pm–deltai;
    last_update=now;
}
```

该算法非常简单,只有一个单一的标记概率 p_m,当缓冲区溢出丢失包的时间间隔大于 freeze_time$d$ 时,增加 p_m,而链路空闲时间间隔大于 freeze_time$i$ 时减少 p_m,而 delta$d$ 和 delta$i$ 为增加和减少的步长,在实际实现时可以考虑将包丢失和链路空闲事件取不同的值.

1.2 EBLUE机制

IETF 的 INTSERV 工作组负责定义关于集成服务的相关协议和标准,为了支持 controlled-load 服务,必须在源主机或网络接入点有相应的侦察(policing)和标记(marking)处理,使进入网络的流量符合通过 RSVP 预留的带宽.

EBLUE 没有使用 RSVP 信令,而是在网络入口点(源主机或 Intranet 和 Internet 的接口处)根据预先协商的流量说明 T_{spec} 对连接进行侦察和标记*处理,然后在路由器根据包是否标记进行区分处理.T_{spec} 包括以下参数:r_m 说明该连接的平均速率,r_p 说明其峰值速率,b 是应用生成的最大突发块大小,此外还包括包的最大和最小长度.

使用令牌桶在网络入口点侦察流量,令牌生成进程根据源的 T_{spec} 生成令牌,平均生成速率为 t_m,短期峰值速

* 本文涉及两种标记,一种是在端主机的优先级标记,我们通过优先级标记可以支持带宽预留;另外一种标记指的是在路由器进行的 ECN 标记,通过它来通知源网络发生了拥塞.当没有明确指明是什么标记时,我们暗指优先级标记.

率为 t_p,令牌桶的深度为 b,每当有包被注入网络时,如果有相应数目的令牌则消耗这些令牌,这些包称之为顺从流(conformant traffic),并对这些包进行标记,否则归入非顺从流(nonconformant traffic),不标记这些包.

路由器除了标记区分处理外,一般还需有接入控制机制,以便预留的聚合流量带宽不会超过路由器的容量,因为本文的主要目的不是讨论接入控制,故假设聚合流量预留级别不超过路由器的容量.

为了能在路由器处理不同优先级的包,可以采用基于类(class-based queue)的分离队列,然后根据队列优先级进行处理,为了简化队列处理,我们采用 FIFO 队列,队列调度采用加强的 BLUE 算法——EBLUE.为了支持最小速率保证,对 BLUE 算法进行了改进,增加了一个参数 $\max_{th}$,表示在接收未标记包即非优先级包时,EBLUE 队列可达到的最大长度,EBLUE 区分标记和未标记的包,当队列长度超过 $\max_{th}$ 时,则丢弃所有进来的未标记包,对标记包只有队列溢出时才丢弃.其标记处理算法见算法 2,Q_{len} 为当前队列长度,Q_{limit} 为允许达到的最大队列长度,即队列限制.

算法 **2**. EBLUE 优先级标记处理算法.

```
if (包未标记){
 if (Q_len>max_th)
       丢弃该包;
 else {
       if (源与路由器支持 ECN){
             以概率 P_m 进行拥塞标记;
             转发该包;
       }else
             以概率 P_m 丢弃该包;
 }
}
if (包被标记){
   if (Q_len<Q_limit)
        转发该包;
   else
        丢弃该包;}
```

为最小化标记包的丢弃,要合理地设置 $\max_{th}$,假设有 n 个有最小速率保证的连接,其峰值速率分别为 r_p^i,服务速率为 L,缓冲区队列限制为 Q_{limit},未标记包的数目为 $\max_{th}(\text{umarked})$,若要保证没有标记包丢弃,则要满足如下条件:

$$\left(\sum_{i=1}^{n} r_p^i - L\right)\times\left(\max_{th}(\text{umarked})/L\right) < Q_{\text{limit}} - \max_{th}(\text{umarked}) . \tag{1}$$

在路由器队列中未标记包的最大数目为 $\max_{th}(\text{umarked})$,路由器要花费 $\max_{th}(\text{umarked})/L$ 的时间才能转发完这些包,而标记包以速率($\sum_{i=1}^{n} r_p^i - L$)递增,因此上述公式说明,如果在时间 $\max_{th}(\text{umarkded})/L$ 内增加的标记包数目小于队列剩余的空间 $Q_{\text{limit}}-\max_{th}$,则保证没有标记包丢弃.但分析该不等式我们知道,要所有的源同时突发地以峰值速率发送数据的概率是很少的,并且 controlled-load 服务也不苛求顺从流包的零丢失,因此我们可以适当放松上述条件限制,根据经验和统计特性,如采用基于度量的接入控制机制[6],可以减小 Q_{limit} 和 L,尽可能保持链路的高利用率和低的丢包率.

1.3 EBLUE与RED的实验结果比较

我们在 ns[7]仿真环境下实现了 EBLUE 算法,实验采用的网络拓扑如图 1 所示,实验结果比较如图 2~图 4 所示.

图 3 和图 4 表示在图 1 所示的网络拓扑情况下,在瓶颈点分别采用 RED 和 EBLUE 算法时丢包概率和队列长度的比较,每隔 30 秒钟增加 400 个 TCP 连接,数据采样时间间隔为 500ms,队列限制为 200KB,源和路由器都

支持ECN.EBLUE算法参数为max_{th}*是180KB,freeze_time*d*和freeze_time*i*均为50ms,delta*d*和delta*i*均为0.01,p_m初始值为0;RED的参数为min_{th}为60KB,max_{th}为180KB,max_p为0.05.从图3我们可以看到,RED随着连接数目的增加丢包率相应增加,并且丢包率较大,而EBLUE算法在所有情况下丢包率基本上为0,只是在突然增加连接数目时,由于p_m尚未来得及进行相应调整,因而丢包率不为0.从图4中可以看出,RED的队列长度基本上保持max_{th},经常超过队列限制而丢包,结合图2可知,EBLUE自动调节ECN标记概率p_m而保持队列长度适中,因而不会超过队列长度限制而导致包丢失.因此EBLUE完全保持了BLUE算法的低丢包率、队列长度适中的特性.

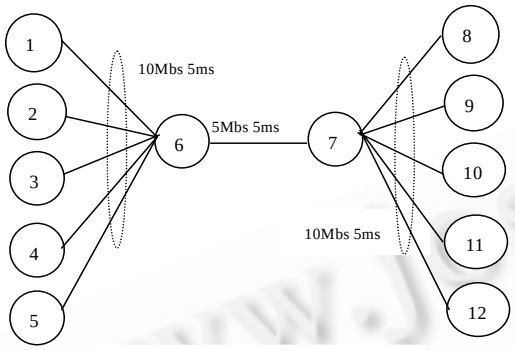

Fig.1 Network topology

图1 网络拓扑图

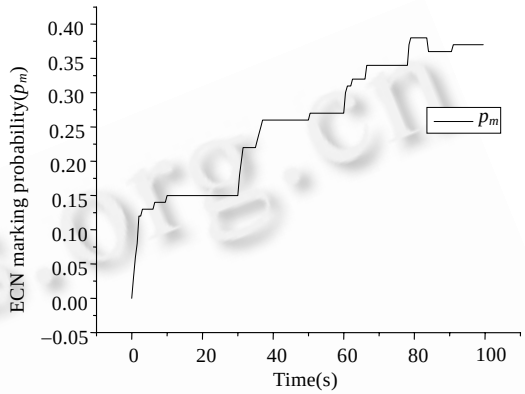

Fig.2 ECN marking probability(p_m)

图2 ECN标记概率(p_m)

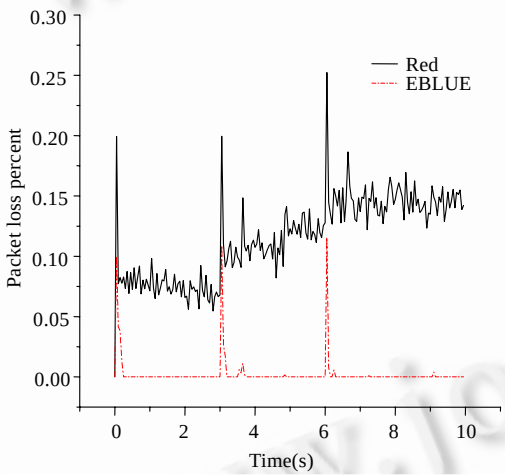

Fig.3 Packet loss percent of EBLUE and RED

图3 EBLUE与RED丢包概率比较

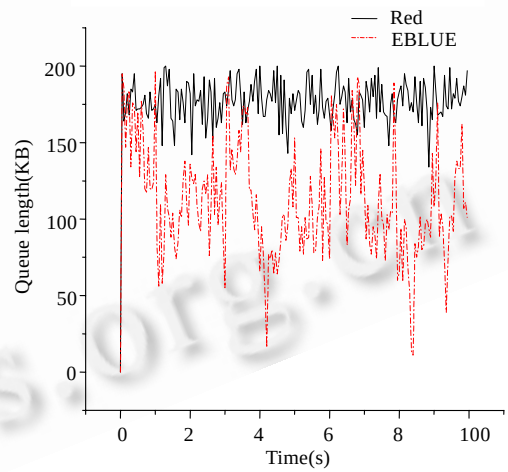

Fig.4 Comprison of queue length of EBLUE and RED

图4 EBLUE与RED队列长度比较

2 支持最小速率保证的TCP协议

文献[8]研究了TCP协议在改进的RED路由器调度算法下支持最小速率保证的TCP拥塞控制机制,本文将结合EBLUE,利用令牌桶标记研究TCP协议支持最小速率保证的机制.

* RED队列管理算法要求max_{th}不应过大,但为了比较RED和EBLUE在连接数目较多、max_{th}较大时的丢包率,特设置max_{th}接近队列限制,RED基于平均队列长度的队列管理算法导致了严重的丢包,而EBLUE仍然能保持基本上为0的丢包率(具体处理算法请见图2及相应的说明).

2.1 TCP拥塞控制算法改进

传统的 TCP 拥塞控制由发送者可同时发送未确认的最大数目包的拥塞窗口 CWND(congestion window),接收者可同时接收最大数目包的窗口 RWND 以及一些与慢速启动、拥塞避免、快速恢复等有关的参数组成.

为了更好地将 TCP 的拥塞控制、令牌桶标记、EBLUE 结合以提供带宽预留服务,减少 TCP 拥塞窗口的波动,文献[8]将 TCP 的拥塞窗口分成两部分:相应预留带宽的部分(rCWND,预留速率与估计 RTT 的乘积再除以 TCP 数据段平均大小)以及与其他连接共享剩余带宽的可变尽力传送部分(CWND–rCWND).

这样,将拥塞窗口分成两个部分之后,TCP 的拥塞控制算法必须作相应的改动,算法如下:

算法 **3**. 改进的 TCP 拥塞窗口更新算法.

(1) 每收到一个新的数据段应答

```
if (CWND<SSTHRESH)
    CWND=CWND+(CWND–rCWND)/CWND;
else
    CWND=CWND+1/CWND;
```

(2) 当重复应答(NDUP)超过某个值(如(3))时

```
CWND=rCWND+(CWND–rCWND)/2+NDUP;
SSTHRESH=rCWND+(CWND–rCWND)/2;
```

(3) 检测到重传超时

```
CWND=rCWND+1;
SSTHRESH=rCWND+(CWND–rCWND)/2;
```

在算法 3 的(2)、(3)两种情况下,临界值(SSTHRESH)都更新为 rCWND+(CWND−rCWND)/2 和 RWND 的最小值,而不是 rCWND+1 和 RWND 的最小值,另外一个重要的改变是,CWND 的更新不同于传统的 TCP 算法.在实际实现时,要求接收方窗口(RWND)至少要等于 rCWND.

算法 3 假设预留速率和小于瓶颈带宽,路由器和源没有采用任何接入控制机制,当网络严重拥塞时,一方面可采用基于度量的接入控制机制处理连接的建立,另外,需要在发送端检测丢失的包是否标记包,如果标记包丢失,则同样要减少 rCWND,这样可以避免和控制网络拥塞.

2.2 令牌标记

在源端采用令牌桶对注入网络的流量进行标记,但若令牌桶的深度过大,将导致过多的突发数据量进入网络,因而只允许少量的连接数目.为了保持较小的令牌桶,同时又能有效地支持预留带宽,文献[8]在发送方采用定时传输机制,算法如下:

算法 **4**. 基于定时器的传输算法.

(1) 每收到一个新的数据段应答

```
if (未确认的包数目小于 CWND 和 RWND)
  if (可用的令牌数目>包大小)
      将包标记后发送;
  else
      将包作为非标记包发送;
```

(2) 定时器过时事件

```
if (未确认的包数目小于 RWND)
  if (可用的令牌数目>包大小)
      将包标记后发送;
复位定时器;
```

算法 4 实际上并没有改变传统的基于应答触发的传输机制,只是每个预留带宽的连接增加了一个定时器,

一旦定时器过时,并且有足够可用的令牌数目,则暂时忽略拥塞窗口,将包作为标记包发送.之所以可以忽略拥塞窗口,是因为当令牌桶有空闲令牌时,则说明接入控制合同允许发送方注入新的顺从标记流量到网络,但所发送的未确认的总的包数目必须小于接收方窗口,否则会导致接收方缓冲区溢出.

为了保证不致丢失令牌,定时器间隔要小于等于[(桶大小–(包大小–1))/令牌生成速率],这样即使在最坏情况下,每次在定时器过时发送完一个包后,令牌桶中的令牌数目最多只有(包大小–1)个.

2.3 实验结果分析

为了验证 EBLUE 和支持最小速率保证的 TCP 协议,我们在 ns2.1b5[7]实现了上述算法,网络拓扑如图 1 所示,实验结果如图 5 所示.

EBLUE 参数为:队列限制设置为 120KB,$\max_{th}$ 是 100KB,freeze_time*d* 和 freeze_time*i* 均为 100ms,delta*d* 和 delta*i* 均为 0.01,p_m 初始值为 0.定时器间隔设置为 20ms,令牌桶深度设置为 50ms 内预留速率所能传输的数据量.

我们实验了两个预留带宽的 TCP 连接和两个尽力传送的 TCP 连接竞争瓶颈带宽的情况,结果显示,有预留的连接不但能获得其预留的带宽,而且能与尽力传送流公平地竞争剩余的带宽.

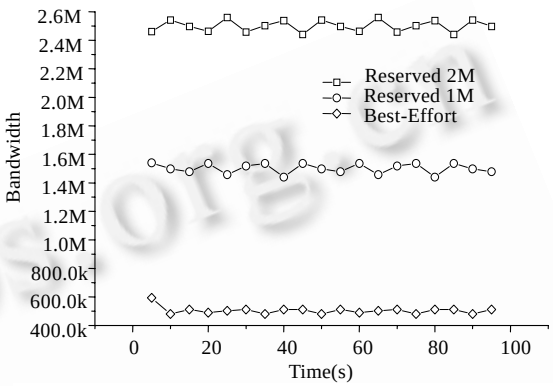

Fig.5 TCP throughput for EBLUE

图 5 TCP 流吞吐率比较(采用 EBLUE 队列机制)

3 结束语

本文提出了一种改进的 BLUE 主动队列管理机制——EBLUE,它不但能简化队列管理,而且能有效地消除包丢失,克服了传统的队列管理机制根据队列长度估计网络拥塞状态所存在的缺点.然后结合 EBLUE 和令牌标记,研究了支持最小带宽保证的 TCP 控制机制,仿真实验结果证明,EBLUE 能有效地支持带宽预留服务.

随着基于 UDP,RTP 的实时连续多媒体体流量的增加,我们下一步的目标是研究 TCP 友好的基于 RTP 的端端拥塞控制,然后研究 EBLUE 对非 TCP 流量(如 RTP)的带宽预留服务支持.

References:

[1] Floyd, S. TCP and explicit congestion notification. Computer Communication Review, 1994,24(5):10~23.

[2] Ramakrishnan, K., Floyd, S. A proposal to add explicit congestion notification (ECN) to IP. Technical Report, RFC 2481, 1999.

[3] Braden, R., Clark, D., Crowcroft, J., *et al*. Recommendations on queue management and congetion avoidance in Internet. Technical Report, RFC2309, 1998.

[4] Feng, W. Improving Internet congestion and queue management algorithm [Ph.D. Thesis]. University of Michigan, 1999.

[5] Wroclawski, J. Specification of controlled-load network element service. Technical Report, RFC 2211, MIT, Cambridge, MA, 1997.

[6] Jamin, S., Danzig, P., Shenker, S., *et al*. A measurement-based admission control algorithm for integrated services packet networks. In: Keshav, S., ed. Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM'95) on Applications, Technologies, Architecture, and Protocols for Computer Communications. Massachusetts: ACM Press, 1995. 2~13.

[7] McCanne, S., Floyd, S. UCB/LBNL/VINT Network Simulator, 1999. http://www-mash.cs.berkeley.edu/ns.

[8] Feng, W., Kandlur, D. Understanding and improving TCP performance over networks with minimum rate guarantees. IEEE/ACM Transactions on Networks, 1999,7(2):173~186.

An Active Queue Management Mechanism Supporting the Priority Marking*

LI Fang-min[1], YE Cheng-qing[2]

[1](*Department of Computer Science, Xiangtan Polytechnic University, Xiangtan* 411201, *China*);

[2](*Department of Computer Science, Zhejiang University, Hangzhou* 310027, *China*)

E-mail: lifangmin@163.net

http://www.xtpu.edu.cn

Abstract: With the increasing of Internet traffic, the RED (random early detection) algorithm depending on the average queue length has inherent flaws. RED cannot efficiently prevent the packet loss even if it combines with explicit congestion notification (ECN). Based on comparison between RED and BLUE, an enhanced active queue management mechanism, EBLUE (enhanced BLUE), is presented. The TCP congestion control with EBLUE is researched, and the token bucket is used to mark priority traffic at the network entry in order to provide the minimum rate guarantees. Finally, under the NS environment, the simulated experiment and performance evaluation are done in packet loss percent and link efficiency for EBLUE by comparing the mechanism with relative ones. The experimental results show that this mechanism can support bandwidth reservation services efficiently.

Key words: active queue management; congestion control; token bucket marking

* Received February 17, 2000; accepted October 16, 2000

Supported by the National Natural Science Foundation of China under Grant No.69974031