

文章编号:1001-9081(2005)07-1514-03

一种基于 ECN 的带优先级的队首标记拥塞控制算法

李洪春,刘 群,丛延奇,刘骁建

(哈尔滨工程大学,计算机科学与技术学院,黑龙江 哈尔滨 150001)

(lihongchun19@mail.china.com)

摘 要:通过对 RED 算法及 ECN 算法进行研究分析,提出了一种带优先级的队首标记的拥塞控制算法。当网络需要预防拥塞发生时,标记将以最快的速度到达接收端,并且被确认帧带回到发送端,因而能够及时预防拥塞的发生,有效提高网络性能。

关键词:拥塞控制; ECN; 队首标记; 优先级

中图分类号: TP393.07 **文献标识码:** A

Head-signed congestion control algorithm based on ECN and priority

LI Hong-chun, LIU Qun, CONG Yan-qi, LIU Xiao-jian

(College of Computer Science and Technology, Harbin Engineering University, Harbin Heilong Jiang 150001, China)

Abstract: After researching RED and ECN algorithm, a head-signed congestion control algorithm was raised based on ECN and priority. The signal was delivered to the receiver as fast as possible, and was brought to the initiator by the acknowledgement frame. As a result, it can prevent the congestion and enhance the performance of network.

Key words: congestion control; ECN(Explicit Congestion Notification); head-signed; priority

0 引言

随着 Internet 上业务量的增加,避免拥塞崩溃^[1]现象的发生成为一项研究的热点。开始,人们利用中间节点丢弃数据包的机制,使端系统减小发送窗口的大小,从而避免拥塞崩溃现象的发生。人们研究了很多主动队列管理算法,最典型的是随机早期丢弃算法(Random Early Detection, RED)^[2]。

RED 算法根据一定的概率对到达的数据包进行丢弃。丢弃的概率由动态变化的平均队列长度来决定,如果缓冲区比较空闲,队列小于一定的长度,则不进行丢弃。当缓冲区队列达到一定的长度后,表明网络将要出现拥塞现象,则按照计算得到的概率对到来的数据包进行丢弃。当队列达到某一个较大的长度后,丢弃所有到达的数据包。这里采用的队列长度是加权平均队列长度。

当一个数据包在到达目的地之前被丢弃,它在传输过程中消耗的所有资源都被浪费了。并且重新传送后,到达目标的数据包顺序会发生颠倒,这也会影响到接收端的处理速度。

RED 算法描述如下:

当有数据包到达时,采用指数加权移动平均的方法计算平均队列长度: $Avg = (1 - w_q) * Avg + w_q * q$

Avg 为加权平均队列长度, q 为数据包的长度, w_q 为权重。

if $Avg < min_{th}$

数据包进入队列

Else if $min_{th} < Avg < max_{th}$

$p_b = \max_p(Avg - min_{th}) / (max_{th} - min_{th})$

$p_a = p_b / (1 - count * p_b)$

以概率 p_a 丢弃到达的数据包

Else if $max_{th} < Avg$

丢弃到达的数据包
Count 为从上次丢弃数据包后成功传送的包数

对上述的算法进行分析可以看到,我们的目标是充分利用网络资源,快速成功地把数据发送到目标端点。为了达到这一目的,需要半路丢弃一定数量的数据包。这种丢弃同时又带来一定的负面影响,降低了带宽的利用率,造成数据包的重传,增加网络的负担。而事实上,有些数据包的丢弃是不必要的。于是又出现了 ECN(Explicit Congestion Notification)算法。

1 ECN 算法及其改进的必要

IETF 在文献[3]中规定了 IP 头的 TOS 域中的第 6、7 位作为 ECN 表示域。在 RED 算法中,发送端通过有包被丢弃知道网络发生了拥塞。如果接收到了三个重复的确认,则发送端快速重传数据包,否则必须等到重传计时器超时才能重传,这一般需要两个 RTT 的时间。在 ECN 算法^[4]中,中间节点在收到一个数据包后,采用和 RED 相同的方法计算队列的加权队列平均长度,然后和 min_{th} , max_{th} 进行比较。如果 Avg 在二者之间,则用与 RED 同样的方法计算概率 p_a ,如果 $p_a > 1$,并不是将其丢弃,而是对其 CE 位进行标记,然后排入队列进行转发。如果 Avg 大于 max_{th} 标记到达的全部数据包。在接收端,如果收到的数据包已经被标记过,则在发送的 ACK 中把这个标记捎带回去。发送端收到了带有标记的 ACK 后,采用和 RED 中包被丢弃相同的方法减小发送速率。

通过这两种主动队列管理方法进行比较我们发现,ECN 算法能够减少不必要的丢包数量,并能让发送端得到显示的拥塞通知。当接收到带有标记的 ACK 后,端点立刻就能判断出网络中发生了拥塞,并迅速减小发送窗口的大小,从而有效地提高网络的反应速度,避免拥塞崩溃现象的发生,提高了网

收稿日期:2004-12-16; 修订日期:2005-04-20

作者简介:李洪春(1980-),男,辽宁朝阳人,硕士研究生,主要研究方向:计算机网络、并行算法、网络计算; 刘群(1957-),男,黑龙江哈尔滨人,教授,博士生导师,主要研究方向:人工智能、信息融合、软件工程; 丛延奇(1964-),男,黑龙江哈尔滨人,副教授,主要研究方向:计算机网络、并行算法与网络计算、无线通信; 刘骁建(1980-),男,河南南阳人,硕士研究生,主要研究方向:数据库、知识库。

络的稳定性。

虽然ECN算法在性能上比RED有所提高,但是在普通的ECN算法中,发送端点得到拥塞通知的提前时间是有限的。网络对拥塞的反应速度提高的也非常有限。为此,我们对ECN进行改进,以明显提高网络的反应速度,有效控制网络的拥塞。

2 基于ECN带优先级的队首标记算法

2.1 算法思想

本算法目的在于让发送端尽快接收到网络拥塞的通知,及时减小发送窗口的大小,从而避免拥塞崩溃现象的发生。

2.1.1 优先传送

我们在中间节点的每个发送端口处增加一个高优先级的队列,称为特权队列。如果特权队列中有数据包,总是首先发送其中的数据包,然后再发送普通队列中的数据包。特权队列中的数据包是接收到的已经被标记过的数据包。这就保证了带有拥塞标记的数据包能够比普通数据包优先转发出去,从而快速传递到端系统中。

2.1.2 队首标记

在普通的ECN算法中我们对新到达的数据包按照计算得到的概率进行标记处理,然后把它排到队尾等待转发。由于标记处的节点正是网络的瓶颈,所以排队所用的时间比重比较大,这就使发送端不能及时接收到拥塞的标记。而且这种时间延迟又是随机的,无规律的。为了尽量避免或减少这种延迟,我们采用队首标记法进行处理。

为每个端口设置标记计数器 $sign$,用来表示需要向网络中发送的标记数量。当网络中接收到一个数据包时,计算标记概率 p_a ,如果 $p_a > 1$,我们并不是直接对数据包进行标记,而是把变量 $sign$ 加1,表示需要向端点发送一个拥塞标记;否则直接排到队列末尾。当节点发送普通队列中的数据包时,首先查看 $sign$ 是否大于1,如果大于1,说明需要向端点发送拥塞标记,那么对要发送的数据包进行标记,同时把 $sign$ 减小1。如果 $sign$ 等于0,则表示网络状况良好,不需要向端点发送拥塞标记,那么就直接发送数据包而不作标记。

2.2 算法描述

接收数据包时:

计算 Avg ;

if $Avg < min_{th}$ && CE 位没有标记

数据包进入普通队列;

else if $Avg < min_{th}$ && CE 位有标记

数据包进入特权队列;

else if $min_{th} \leq Avg < max_{th}$

{

计算 p_a

if $p_a < 1$ && CE 位没有标记

数据包进入普通队列;

else if $p_a \geq 1$ && CE 位没有标记

{ $sign = sign + 1$;

数据包进入普通队列; }

else if $p_a \geq 1$ && CE 位有标记

{ $sign = sign + 1$;

数据包进入特权队列; }

else if $p_a < 1$ && CE 位有标记

数据包进入特权队列;

}

else if $Avg \geq max_{th}$ && CE 位没有标记

{ $sign = sign + 1$;

数据包进入普通队列; }

else if $Avg \geq max_{th}$ && CE 位有标记

{ $sign = sign + 1$;

数据包进入特权队列; }

发送数据包时:

if 特权队列非空

从其中取出一个进行转发;

else

{ 从普通队列中取出一个数据包;

if $sign > 0$

{ 对数据包作标记,并发送出去;

$sign = sign - 1$;

else

直接发送数据包;

}

2.3 算法分析

在瓶颈带宽处,如果经过计算需要给端点一个拥塞标记,到达的数据包虽然排入队列末尾,但是标记却被马上发送的队首数据包传递出去了。而且一旦数据包被标记了,以后走的就是特权队列了,延迟非常小。相对于普通的ECN算法来说,标记能提前到达发送端系统,从而能提高网络的反应速度,有效避免拥塞崩溃现象的发生,减少了不必要的数据包丢失概率,有效利用了网络带宽。这种算法的特点是并不需要中间节点进行大量的计算,没有带来过多的负担,实用性强,容易实现。

3 仿真结果

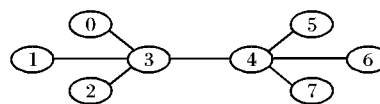
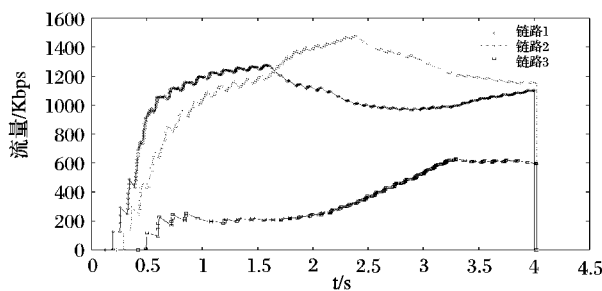
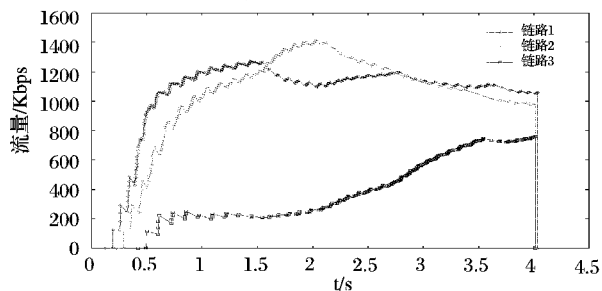


图1 拓扑结构



(a) RED/ECN流量



(b) HPECN流量

图2 流量对比

本仿真实验在网络仿真软件 ns2^[5] 下进行,采用的网络拓扑结构如图1。

其中,n0到n5,n1到n6,n2到n7是3条TCP连接,n0,

n_1, n_2 是发送端, n_5, n_6, n_7 是接收端。 n_3 到 n_4 是瓶颈链路, 带宽为 3m, 时延为 10ms, 队列长度为 20。 在本改进的算法中, 瓶颈链路的普通队列长度为 19, 特权队列长度为 1, 其余属性不变。 其余链路都是带宽为 3m, 时延为 10ms 的链路。 其中 n_0 到 n_5 之间的连接 1 在 0.1s 开始传输数据, n_1 到 n_6 之间的连接 2 在 0.2s 开始传输数据, n_2 到 n_7 之间的连接 3 在 0.4s 开始传输数据。 3 条连接在 4.0s 同时停止传送。

图 2(a) 和 (b) 分别为改进前和改进后不同算法下测量得到的 3 条链路的流量对比, 通过对比发现, 瓶颈链路处的总体流量有所增加, 另外, 在 HPECN 算法中链路的公平性也有所提高。

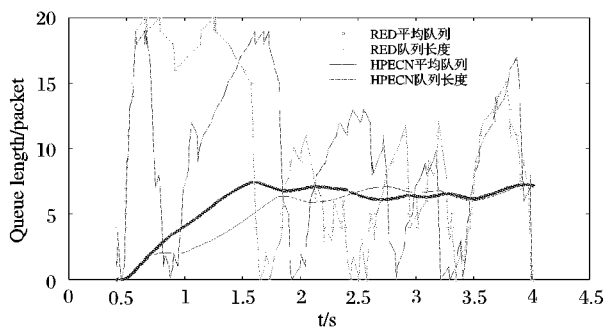


图3 队列长度对比

图 3 是改进前后队列长度的对比, 细线是改进后的平均队列长度。 通过对比可以发现, 经过改进后的平均队列长度整体上降低了。 这主要是因为发送端及时收到了拥塞控制信

(上接第 1513 页)

负载变化造成的。

图 2 为在不同的更新周期 T_p 下, 各调度算法中任务的平均响应时间。 其中: random 为随机算法; shortest 为最小负载调度算法; $k=2$ 为 k 子集算法; RLBA 为本算法。 shortest 算法和 k 子集算法都只考虑了 CPU 资源。 实验结果表明, RLBA 算法在避免调度饥饿的同时, 轻载节点被命中的概率要高于 k 子集算法, 被调度节点上的资源相对宽裕, 会更适合任务的运行。 改变 S 可以划分更多的节点类型, 调整 $C[S+1]$ 内容可以改变某一节点所属的节点类型, 改变 b_k 可以扩大或者缩小调度候选节点的范围。

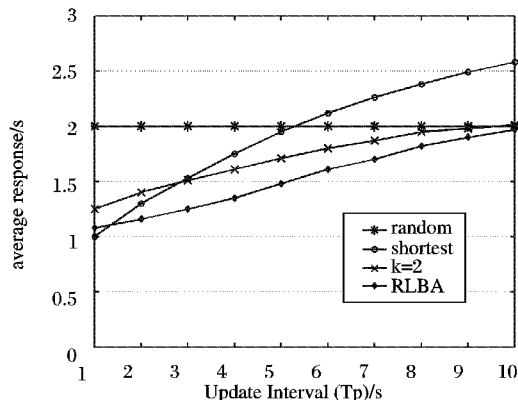


图2 各算法任务平均响应时间

4 结语

RLBA 算法的基本思想是根据任务对系统资源的利用特点, 将其分派到一个资源相对宽裕的节点上。 它结合了 k 子

息, 从而在带宽相同的情况下有效控制了队列的长度, 预防了拥塞崩溃现象的发生。

另外, 从仿真结果的数据分析得到, 在普通的 RED 算法中, 丢包率高达 1.207%, 而在带优先级的队首标记 ECN 算法中丢包率和 ECN 中相同, 为 0.452%, 明显降低了很多。

4 结语

本算法通过引进了队首标记和优先传递的策略, 有效改进了网络的性能, 在相同的网络资源条件下, 能够适度提高网络的流量, 降低网络的数据包丢失概率。 但还存在一些问题, 比如增加了接收端到来的数据包序号提前的可能性, 在系统开始阶段还不能有效控制数据包的丢失, 不能非常有效地解决公平性等问题。

参考文献:

- [1] NAGLE J. RFC 896, Congestion Control in IP/TCP[S]. IETF, 1984.
- [2] FLOYD S, JACOBSON V. Random early detection gateways for congestion avoidance[J]. IEEE/ACM Transactions on Networking, 1993, 1(4): 397-413.
- [3] NICHOLS K, BLAKE S, BAKER F, et al. RFC 2474, Definition of the differentiated services field (DS field) in the IPv4 and Ipv6 headers[S], 1998.
- [4] RAMAKRISHNAN K. RFC 3168, The addition of explicit congestion notification (ECN) to IP[S], 2001.
- [5] MCCANNIE S, FLORD S. ns - LBNL the network simulator[EB/OL]. <http://www.isi.edu/nsnam/ns>, 2004-12.

集算法^[1,2]和阈值算法^[7]的优点, 具有选择范围动态变化和低负载节点优先命中的特征。 实验表明, RLBA 算法具有较好的性能。

在建立异构系统的负载描述模型后, 本算法可以很容易扩展到异构系统中。

参考文献:

- [1] MITZENMACHER M. The Power of Two Choices in Randomized Load Balancing[J]. IEEE Transactions on Parallel and Distributed Systems, 2001, 12(10): 1094-1104.
- [2] DAHLIN M. Interpreting Stale Load Information[J]. IEEE Transactions on Parallel and Distributed Systems, 2000, 11(10): 1033-1047.
- [3] WANG YB, HYATT R. An improved algorithm of two choices in randomized dynamic load-balancing[A]. Proceedings of the fifth international conference on algorithms and architectures for parallel processing [C], 2002.
- [4] 唐丹, 金海, 张永坤. 集群动态负载均衡系统的性能评价[J]. 计算机学报, 2004, 27(6): 803-811.
- [5] 蒋江, 张民选, 廖湘科. 基于多种资源的负载均衡算法的研究[J]. 电子学报, 2002, 30(8): 1148-1152.
- [6] AMIR Y, AWERBUCH B, BORGSTROM R, et al. An Opportunity Cost Approach for Job Assignment in a Scalable Computing Cluster[J]. IEEE Transactions on Parallel and Distributed Systems, 2000, 11(7): 760-767.
- [7] 周佳祥, 郑伟民, 杨广文. 一种基于进程迁移的自适应双阈值动态负载均衡系统[J]. 清华大学学报(自然科学版), 2000, 40(3): 121-125.