一种基于速率的组播拥塞控制算法及其性能分析

苏晓丽[1],郑明春[2],李锦涛[1],孟 强[2]
(1. 中国科学院计算技术研究所数字化室,北京 100080;2. 山东师范大学计算机科学系,山东济南 250014)

摘 要: 大部分组播拥塞控制机制都是将包丢失作为网络拥塞的信号,存在丢包、响应速度慢以及由此引起的协议间不公平等缺陷. 本文提出了一种新的基于速率拥塞控制算法通过对拥塞的早期检测,进行及时反馈,发送端通过调节数据包的发送间隔进行拥塞避免和控制,使网络能够对拥塞做出快速反应,更有效地利用网络资源. 实验结果表明,在相同的配置下,采用该拥塞控制算法的网络在吞吐量、灵敏性和公平性等性能上均优于原先的算法.

关键词: 组播; 拥塞控制; TCP 友好; RBMCC
中图分类号: TP393 **文献标识码**: A **文章编号**: 0372-2112 (2004) 02-0330-05

A Rate-Based Multicast Congestion Control Algorithm and Its Performance Analysis

SU Xiao-li[1], ZHENG Ming-chun[2], LI Jin-tao[1], Meng Qiang[2]
(1. *Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100080, *China*;
2. *Department of Computer Science, Shandong Normal University, Jinan, Shandong* 250014, *China*)

Abstract: A novel rate-based multicast congestion control scheme —RBMCC is presented, which uses the positive explicit congestion indication and proxy-based feedback control scheme to inform the sender about the status of network, so that the sender can detect incipient congestion early and adjust the send rate to alleviate congestion by changing packet intervals. The mechanism was implemented in the Network Simulator and its performance was compared with other algorithms by simulation. The simulation results show that RBMCC improves the dynamics and responsiveness, uses the available resource of network efficiently, and also has good TCP-friendliness.

Key words: multicast; congestion control; TCP-friendly; RBMCC

1 引言

随着计算机和网络技术的发展,越来越多的新型应用需要组播协议的支持以节省网络带宽,提高传输效率,如信息发布,软件分发,web 代理更新和同步数据恢复和更新等. 然而由于没有良好的拥塞控制机制,组播在 Internet 上的应用受到了极大的限制,IETF 指出在标准化可靠组播协议之前拥塞控制是必须被解决的一个问题[1]. 因为,今天 Internet 的成功很大程度上得益于 TCP 拥塞控制算法的引进[2],它确保了网络的稳定性,防止了拥塞崩溃[3]. 但组播使用的是用户数据报协议(UDP),UDP 是一种"尽力而为"(Best-effort)协议,没有内建的拥塞控制机制. 组播流以一种不公平的方式与 TCP 流竞争:当遇到拥塞时,所有参与的 TCP 流减小,速度试图减轻拥塞,而非 TCP 流继续以原速发送,这种极度不公平的情形会遏制 TCP 通信,甚至导致拥塞崩溃. 由于 TCP 通信占整个 Internet 通信的 90 %以上,组播应用要想成功就必须开发出能与 TCP 友好共存的组播拥塞控制机制. 目前主要通过两种方式来保证 TCP 友好性:基于窗口的拥塞控制和基于速率的拥塞控制. 基于窗口的拥塞控制主要是模仿 TCP 基于窗口的 AIMD 机制来进行拥塞控制[4,5],但这些算法很难解决源端的反馈内陷和组播吞吐量趋于零(drop to zero)等问题[6,7]. 因此许多可靠组播拥塞控制研究都采用基于速率的机制来调节通信,即发送者通过控制数据包的平均传输速度来进行拥塞控制[7,8].

目前的基于速率的拥塞控制机制中主要存在以下几个问题:

首先,大多算法中采用在文献[9,10]给出的两个 TCP 吞吐量公式来计算速率,要求每个接收者精确估计 RTT,丢失率等参数,为了防止随机的丢失的影响,算法采用指数级加权平均方法计算一段间隔内(通常是几个 RTT 时间)的平均丢失率,造成了很大时延,影响了拥塞算法的动态性,使源端不能及时了解网络的拥塞状态. 而 TCP 是一种快速反应的拥塞控制算法,发送端能及时了解网络的状态,可以在一个 RTT 时间内对拥塞做出快速反应,这是 TCP 成功的关键因素,但也使非 TCP 友好的协议能迅速消耗可利用带宽,遏制了 TCP 通

收稿日期:2002-12-02;修回日期:2003-11-17
基金项目:山东省优秀中青年科学家科研奖励基金

信,造成协议间的不公平现象.

其次,组播树广泛分布于整个 Internet,源端必须考虑多条路径上的拥塞状态.但目前大部分单速率拥塞控制机制都是根据 worst-path 公平性,发送者仅对来自最拥塞路径上的丢失指示做出反应,忽略其他接收者的信号,虽然避免了反馈内陷问题,但所有组成员受瓶颈接收者的速度的影响,而且当网络中其他部分发生了更严重的拥塞时,源端需要一定的时间来检测和定位新的拥塞路径,这段时间内源端很难做出正确的反应,造成了网络吞吐量不必要的振荡.

另外,现有的算法中根据包丢失来判断拥塞,将 NACK (Negative Acknowledgement) 作为拥塞反馈信号,源端根据 NACK 来进行拥塞控制,收到 NACK 时降低速度.然而只有当接收者检测到数据包丢失时,才发送 NACK 要求重传,而当网络拥塞减轻,又有资源可以利用时,源端不能及时收到任何反馈来调节速率,只有等到定时器超时,才能进一步增加速度,因此不能有效地利用网络资源.

针对上述问题,本文提出了一种新的基于速率的组播拥塞控制机制 RBMCC(a rate-based multicast congestion control),由接收者代表根据延迟和带宽乘积来检测网络中数据的排队情况来判断拥塞状态,积极地反馈拥塞控制信号 CC(Congestion Control),源端根据来自指定接收者的 CC 信号来进行速率调节而不是等到数据包丢失后才进行拥塞控制,这样可以使源端及时了解网络的状态,对网络条件的变化做出灵敏反应,改善了协议间的公平性,提高了组播会话的吞吐量,算法的性能在网络模拟器 NS2[11] (Network Simulator 2) 上得到了进一步的证明.

2　算法设计

2.1　拥塞检测

为了提高组播拥塞算法的灵敏性,减少包丢失,RBMCC 通过以下两种方式来检测网络的拥塞状态:

2.1.1　根据网络队列延迟检测拥塞

在没有发生丢包现象时,通过监测网络中队列延迟来判断网络拥塞状况,而不是等到丢包时才通知源端进行控制,尽可能避免拥塞的发生,同时减小丢包率.队列延迟按如下方法测量,如图 1 所示:$delay = R\_inv - S\_inv$,S_inv 和 R_inv 表示连续数据包的发送间隔和到达接收方的时间间隔,$delay$ 即数据包在网络中的排队延迟.我们使用 $delay$ 作为反应网络状态的指标,接收者观察 $delay$ 的变化,向源端反馈速率调节状态 r_status,从而使源端能及时地了解网络状态的变化,尽早避免拥塞的发生,其中 S_inv 的测量在 2.3 节算法中实现.

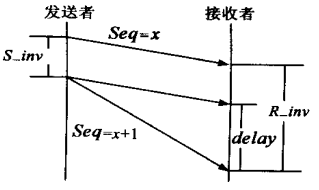

图 1　网络队列延迟

TCP Vegas[12] 中指出数据包排队长度保持在 1～3 之间时,网络性能比较稳定,因此本文采用如下算法:

当 $1 * S\_inv \le delay \le 3 * S\_inv$ 时,通知发送方速率保持不变,如果 $delay < 1 * S\_inv$,则通知发送方增加速率,如果 $delay > 3 * S\_inv$ 则通知发送方降低速率.

2.1.2　根据数据包丢失检测拥塞

虽然通过监视网络队列延迟可以在一定程度上避免拥塞,但由于网络上数据流量的突发性,丢包是不可能避免的.当发生丢包时,意味这网络中某部分出现严重堵塞,应该通知源端迅速降低发送速度,减少网络的负载.为了防止随机丢失的影响,我们对丢失报告进行周期性的反馈.

拥塞检测算法伪码如下:

```
if currentseq = lastseq + 1;
{ delay = R_inv —S_inv;
  if delay < 1 * S_inv
     r_status = increase;
else if delay > 3 * S_inv
        r_status = decrease;
else r_status = normal;
}else r_status = loss;
```

2.2　反馈机制

RBMCC 选取组播树中几条比较拥塞的路径上的接收者作为拥塞代表监视网络状态,这些组代表周期性地向源端提供及时的网络拥塞状态的报告 CC.

发送方负责选择代表和记录代表集合,选择代表时依据三个优先级不同的指标:检测到的排队延迟 $D(delay)$、反馈频率 $F(feedback)$ 和丢失率 $L(loss\ rate)$,它们直接反应了通向接收者的链路状况.丢失率大的接收者意味着严重阻塞,它具有最高的优先权作为代表,其次在没有发生丢失的接收者中,检测到的网络队列延迟较长的接收者具有较高的优先权作为代表,对于延迟或丢失率相似的接收者,选择反馈频率较高的接收者作为代表.在开始阶段,代表集合为空,任何提供反馈的接收者都有资格被选为代表,当反馈到来时,源端将第一个反馈的接收者作为代表,利用接收者的 IP 地址来标识,建立初始代表集合.源端组播当前代表集合到整个组,当一个接收者在代表集合通告中检测到自己是代表集合的成员时,则指定本身为一个代表,它开始检测网络延迟并进行周期性反馈,而其他非代表接收者不再进行反馈,只有发现自己的性能指标 D、F、L 比当前代表更高时才发送代表更新报告,申请加入代表集合.如果一个接收者代表收到源端的代表集合更新通知,它不再是代表时,它将回到非代表操作,停止反馈.代表集合成员是在不断变化的,当新的拥塞出现时,新的代表被选择加入代表集合,当代表集合满时,拥塞程度最低的代表从集合中退出,拥塞程度同样也是以上述三个指标来衡量,发送方每次根据代表集合中拥塞程度最高的代表的反馈信息判断网络拥塞的状态.

非代表接收者采用基于定时器的反馈机制,只有在检测到数据包丢失时才发送反馈,每个非代表接收者在发送反馈信息之前,需要等待一段随机时间,如果在等待期间收到来自其他接收者或代表的对同一数据包的反馈信息,则取消自己的反馈,否则当定时器超时,将反馈发送到源端,并以组播的方式发送给其他接收者,同时源端将根据延迟和丢失率等参数对代表集合进行更新.

2.3 速率调节

RBMCC 算法采用和式增加和式减小(AIAD)和和式增加积式减小(AIMD)两种方式来进行速率调节,源端根据来自接收者代表的拥塞状态报告 r_status 来相应地调节速度. TCP 拥塞控制周期是在一个 RTT 时间内完成,由于 TCP 不能保证具有不同 RTT 的连接之间的公平性,它倾向于传输时间短的连接[13],所以组播速率调节也应采用类似 TCP 的时间尺度,才能保证与 TCP 公平竞争带宽. 该算法通过计算接收者代表的平均 RTT 来调节发送间隔来使发送速度随着网络状态的变化自适应地改变,RTT 的计算通过在数据包头部设置时间戳来实现. 假设发送间隔为 S_inv,则发送速率为 $PackSize/S\_inv$.

在没有数据包丢失时,发送方采用和式增加和式减小 AIAD 方式来改变速率,当接收方的拥塞状态报告 $r\_status = normal$ 时,发送速率保持不变;当 $r\_status = Increase$ 时,发送速率线性增加,每个平均 RTT 增加一个数据包;当 $r\_status = decrease$ 时,发送速率线性降低,每个 RTT 减小一个包;当 $r\_status = loss$ 时,表示在网络中发生数据包丢失现象,产生了严重拥塞,采用和式增加积式减小 AIMD 方式调节速率,发送方将发送速率降低了一半. 为了防止大量独立丢失的影响,发送方只对来自代表的拥塞报告进行反应,每个平均 RTT 时间只调节一次,避免过渡遏制吞吐量.

采用如下算法:

$S\_inv = ave\_RTT$; // 初始化

if $r\_status == normal$

$S\_inv(t+1) = S\_inv(t)$;

else if $r\_status == increase$

$S\_\mathrm{inv}(t+1) = S\_\mathrm{inv}(t) * ave\_RTT/(S\_\mathrm{inv}(t) + ave\_RTT)$;

else if $r\_status == decrease$

$S\_\mathrm{inv}(t+1) = S\_\mathrm{inv}(t) * ave\_RTT/(ave\_RTT - 2 * S\_\mathrm{inv}(t))$;

else if $r\_status == loss$

$S\_\mathrm{inv}(t+1) = S\_\mathrm{inv}(t) * 2$;

$S\_\mathrm{rate}(t+1) = Packetsize/S\_\mathrm{inv}(t+1)$;

3 仿真实验

本文在 NS2 上进行实验,通过与 TFMCC 算法[14]的比较来评价算法性能,在前两节实验中采用了著名的哑铃拓扑作为仿真实验的拓扑结构(如图 2 所示),所有源端在瓶颈链路的左边,所有接收者在右边,瓶颈链路的带宽与共享这条链路的流的数目成比例,每个组播组至少有两个接收者,路由器队列采用 RED(Random Early Drop)算法,数据包大小为 1000 字节.

3.1 灵敏性

下面的两组实验模拟了 RBMCC 与 TFMCC 算法对网络可利用带宽变化的自适应过程,采用了如图 2 所示的拓扑结构,一个组播发送者经过瓶颈链路向三个接收者发送数据. 实验设置如表 1:

表 1 灵敏性实验参数设置

实验 1	带 宽	实验 2	带 宽
0～30s	1.0Mb/s	0～60s	1.0Mb/s
30～60s	0.5Mb/s	60～70s	0Mb/s
60～100s	0.75Mb/s	70～110s	1.0Mb/s
		110～140s	0Mb/s
		140～180s	1.0Mb/s

从图 3、图 4 可以看出,由于 RBMCC 采用了积极的拥塞反馈信号,源端能及时了解网络的状态,能迅速地根据网络带宽的变化做出正确的响应,改善了算法的灵敏性,并且有效地利用了网络资源. 而 TFMCC 由于源端没有足够的信息,无法及时了解网络状态的变化,所以总是滞后 9s 左右才能了解网络的状态,源端不能及时做出正确的调节,这不仅影响了算法的动态性,而且从下一节的实验可以看出,这将对算法的公平性也产生较大的影响.

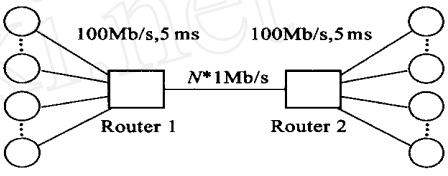

图 2 仿真网络拓扑结构图

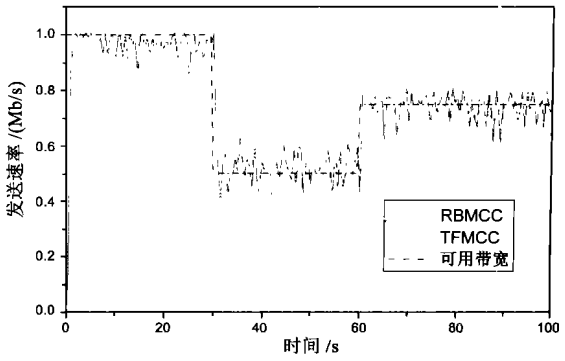

图 3 实验 1 对网络可用带宽的灵敏性比较

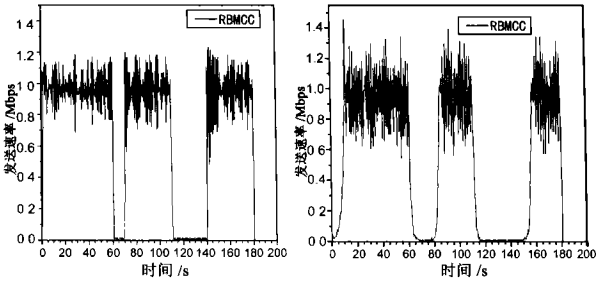

图 4 实验 2 灵敏性比较

3.2 TCP 友好性

为了描述 RBMCC 的性能引入了 TCP 友好性比率的定义,设 K_R, K_T 分别表示 RBMCC 和 TCP 流的数目,RBMCC 流的吞吐量分别为: T_1^R, T_2^R, …, $T_{K_R}^R$, TCP 流的吞吐量分别为 T_1^T, T_2^T,

…, $T_{K_T}^T$,则 RBMCC 流与 TCP 流的平均吞吐量为:

$$T_R = \frac{\sum_{i=1}^{K_R} T_i^R}{K_R} \text{ 和 } T_T = \frac{\sum_{i=1}^{K_T} T_i^T}{K_T}$$

于是可定义 TCP 友好比率为: $F = T_R / T_T$

在实验中采用图 2 所示的拓扑结构,竞争连接的数目在 4 到 40 之间变化,TCP 和 RBMCC 各占一半. 所有源端在 0～5 秒之间选择一个随机时间开始发送,整个模拟过程持续 120 秒. 然后用 TFMCC 代替上面拓扑中的 RBMCC 进行模拟. 实验结果如下:

从图 5 可以看出,RBMCC 流的 TCP 友好性比率在 1 附近小幅度振荡,较之 TFMCC 改善了 TCP 友好性.

3.3　吞吐量

在本节中进行三组实验来观察相关丢失和独立丢失对组播会话吞吐量的影响,采用如图 6 所示拓扑结构,一个发送方通过一棵深度为 3 的树给三个接收者传输数据. 实验 a:共享链路 L1 的丢失率为 10 %,其余链路不发生丢失;实验 b:L1 不发生丢失,链路 L2、L3 经历 10 %的丢失率;实验 c:树的第三层各个不相关链路经历 10 %的丢失率,其余链路不发生丢失. 实验中为了避免队列丢包的影响,设链路带宽应该足够大,如 100Mbps.

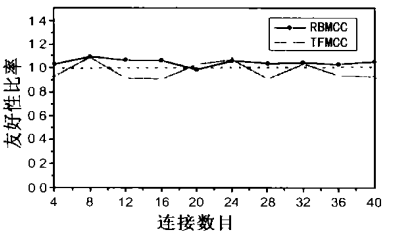

图 5　RBMCC 与 TFMCC 的 TCP 友好性

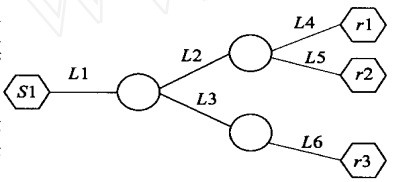

图 6　仿真拓扑结构 2

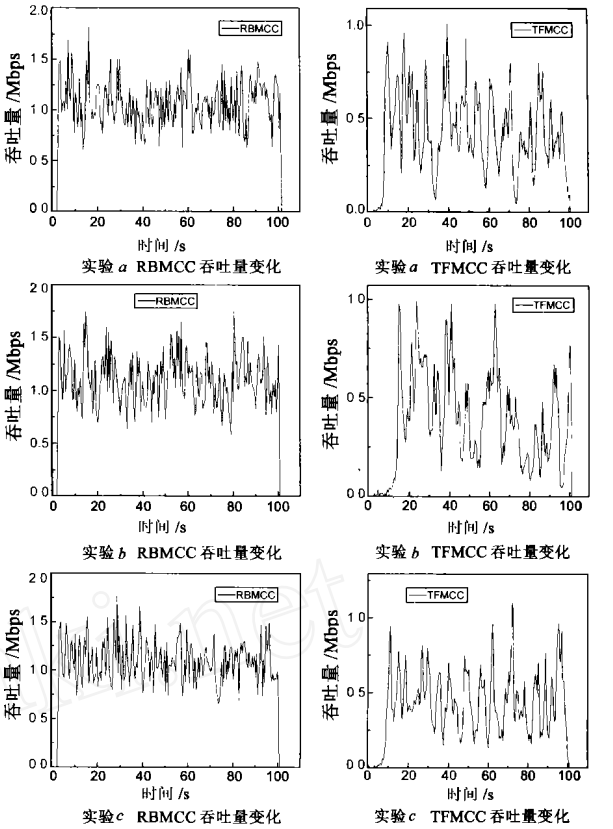

图 7　数据丢失对吞吐量的影响

从图 7 可以看出以下两点:(1) 不管丢失相关(a)或不相关(b、c),三个实验中 RBMCC 都具有相似的平均带宽,受相关丢失和独立丢失的影响较小,这是由于 RBMCC 采用了基于代表集合的积极反馈方式,可以更准确地进行拥塞定位,对整棵组播树有了更全面的了解,源端对网络拥塞变化判断更加准确,能对拥塞做出正确及时的反应;没有过渡扼制吞吐量,具有一定的扩充性. (2) 在相同条件下,由于 RBMCC 同时监控了组播树上的多条拥塞路径,每次根据代表集合来做出判断,一两个接收者拥塞状态变化不会影响源端的正确反应,从而减少了由于单个代表变化引起的盲目反应周期和振荡周期,吞吐量变化幅度较小,而 TFMCC 吞吐量的振荡比较大.

4　结论和进一步的工作

组播拥塞控制已经成为制约组播应用发展的主要因素,本文提出了一种新的基于速率的拥塞控制算法,通过监视网络中的排队延迟,发送积极的拥塞控制信号来使源端能及时了解网络拥塞水平的变化,仿真实验结果表明,该算法改善了灵敏性、公平性等性能,具有较好的拥塞自适应能力. 进一步的研究将侧重于与差错控制相结合的拥塞控制机制以及多速率组播拥塞控制问题.

参考文献:

[1] Mankin A, Romanow A, Bradner S, Paxson V. IETF criteria for evaluating reliable multicast transport and application protocols [S]. IETF RFC2357, June 1998.

[2] Jacobson V. Congestion avoidance and control [A]. Proc of ACM SIGCOMM'88 Symposium [C]. Stanford, CA, USA, 1998. 314 - 332.

[3] Nagle J. Congestion Control in IP/TCP Internetworks [S]. RFC 896, 1984.

[4] Rhee I, Ballaguru N, Rouskas G N. MTCP: Scalable TCP-like Congestion Control for Reliable Multicast [R]. TR-98-01, Department of Computer Science, NCSU, January 1998.

[5] Golestaniand S J, et al. Fundamental observations on multicast congestion control in the internet [A]. IEEE Infocomm'99 [C]. New York, USA, 1999. (1). 990 - 1000.

[6] Bhattacharyya S, Towsley D, Kurose J. The loss path multiplicity problem in multicast congestion control [A]. IEEE Infocom'99 [C]. New York, USA, 1999. 856 - 863.

[7] Whetten B, Conlan J. A Rate Based Congestion Control Scheme for Reliable Multicast [R]. Technical White Paper, GlobalCast Communication, Oct. 1998.

[8] Handley M, Floyd S, Whetten B. Strawman Specification for TCP Friendly (Reliable) Multicast Congestion Control [R]. RMRG Meet-

ing ,Pisa ,Italy ,June 1999.

[9] Mahdavi J ,Floyd S. TCP-Friendly Unicast Rate-Based Flow Control [EB/OL]. Technical note sent to the end2end-interest mailing list , http://www.psc.edu/networking/papers ,1997.

[10] Padhye J ,Firoiu V ,Towsley D ,Kurose J. Modeling TCP throughput :A simple model and its empirical validation [A]. Proc. of SIGCOMM '98 [C]. Vancouver ,CA ,1998. 303 - 314.

[11] UCB/LBNL/VINT Network Simulator - - ns2. http://www.isi.edu/nsnam/ns [CP/OL].

[12] Lawrence S ,Sean W ,Larry L. TCP vegas :New techniques for congestion detection and avoidance [A]. Proc of SIGCOMM '94 [C]. London ,UK ,1994. 24 - 35.

[13] Floyd S ,Jacobson V. Connection with multiple congested gateways in packet-switched networks ,Part 1 :One-way traffic [J]. ACM Computer Communication Review ,1991 ,21(5) :30 - 47.

[14] Widmer J ,Handley M. Extending equation-based congestion control to multicast applications [A]. In Proc of ACM SIGCOMM [C]. San Diego ,CA ,2001. 275 - 286.

作者简介:

苏晓丽 女,1979 年 5 月生于山西省文水县,2003 年获得山东师范大学计算机专业硕士学位,目前为中科院计算所博士研究生,主要研究方向为网络协议、网格技术和网格应用. Email : xlsu @ict. ac. cn.

郑明春 女,1964 年生于山东烟台,山东师范大学教授,硕士生导师,主要研究方向为计算机网络应用,网络拥塞控制等.

李锦涛 男,1962 年 3 月生于湖南,中科院计算技术研究所数字化室,博士生导师,主要研究方向为多媒体网络应用,网络计算,视频编码等.