文章编号:1001 - 893X(2010)09 - 0069 - 04

一种适用于分层多播的 RTT 估算方法*

张 牧[1],王 攀[2]

(1.贵阳学院 计算机科学系,贵阳 550005;2.南京邮电大学 江苏省通信与网络技术工程研究中心,南京 210003)

摘 要:提出了一种新的适用于分层多播拥塞控制方案中的往返时间(RTT)估算方法。该估算方法中,分层多播会话中未被抑制的接收者进行 RTT 测量,被抑制接收者进行单向延时测量来估算 RTT 值,同时该方法采用优先级因子将经常被抑制的接收者和具有较大 RTT 的接收者考虑在内。仿真结果表明该算法是可行的。

关键词:分层多播;拥塞控制;往返时间;估计算法

中图分类号:TP301 **文献标识码**:A doi:10.3969/j.issn.1001 - 893x.2010.09.016

A New RTT Estimation Algorithm for Layered Multicast

ZHANG Mu[1], *WANG Pan*[2]

(1. Department of Computer Science, Guiyang University, Guiyang 550005, China;
2. Jiangsu Province Engineering Research Center of Telecommunication & Network Technology,
Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: A new RTT(Round Trip Time) estimation algorithm for layered multicast congestion control is presented. Non-suppressed receivers in layered multicast session use round trip estimation to get an accurate RTT, suppressed receivers use one-way measurement to get an estimation of RTT, and the algorithm takes the frequently suppressed receivers and receivers with large RTT into consideration using a priority factor. Simulation results show that this algorithm is feasible.

Key words: layered multicast; congestion control; round trip time(RTT); estimation algorithm

1 引 言

异构性是 Internet 的固有特征,接入网络的多样性以及端系统处理能力的差异是这种异构性存在的根源,并且随着 Internet 规模的不断扩大,异构性问题将更加突出。多播的提出为异构网络环境部署多媒体多播应用提供了可行的方案,其中数据分层多播技术以多种速率传输既满足了不同用户间的异构性和不同需求,又避免了冗余数据传输带来的带宽浪费,成为多播研究领域中的一个热点。

目前,传输控制协议(TCP)通信量占 Internet 通信的 90%以上,多播应用的广泛开展引起了非 TCP 通信量所占比重的增加[1]。这些协议大部分没有与 TCP 兼容的拥塞控制机制,它们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 流立即减小其速度试图减轻拥塞,而非 TCP 流继续以原速率发送,这种极度不公平的情形会遏制 TCP 通信,甚至导致拥塞崩溃。当网络中可利用的带宽被耗尽,所有的包在到达目的地之前将会因拥塞而

* **收稿日期**:2010 - 05 - 10;**修回日期**:2010 - 06 - 17
基金项目:国家高技术研究发展计划(863 计划)项目(2006AA01Z232);贵州省科技厅科技支撑(工业攻关)项目((2008)3034);贵阳市人才创新创业资金资助项目;贵阳市科技局信息化项目
Foundation Item: The National High - Tech Research and Development Program of China (863 Program)(No. 2006AA01Z232); Guizhou Province Technology Program(No. (2008)3034); Talent Innovation & Startup Project of Guiyang; The Informatization Project of Guiyang

被丢弃,因而多播应用要想成功地在 Internet 上应用,必须使用能与 TCP 友好共存的多播拥塞控制机制,保证与 TCP 共存时的公平性,即 TCP 友好性。文献[2]提出了一种精确计算 TCP 吞吐量的公式以确保分层多播拥塞控制方案满足 TCP 友好性,该公式显示往返时间(RTT)是计算 TCP 吞吐量的关键因素之一,因此 RTT 的计算对于确保多播会话满足 TCP 友好有着非常重要的作用。

本文提出了一种新的适用于分层多播拥塞控制方案中的 RTT 估算方法。在该估算方法中,分层多播会话中未被抑制的接收者进行 RTT 测量,被抑制接收者进行单向延时测量来估算 RTT 值,同时该方法采用优先级因子将经常被抑制的接收者和具有较大 RTT 的接收者考虑在内。

2 往返时间(RTT)计算方法

目前大多数多媒体应用都是基于 RTP/RTCP(实时传输协议)协议[3]的,考虑到 RTP/RTCP 协议应用的广泛性,我们对 RTCP 协议进行扩充,增加 RTT 的请求和应答字段来进行 RTT 的测量,同时也可以进行单向延时测量。我们采用闭环的方式[4]进行 RTT 测量,重新定义的 RTCP 报文如图 1 所示。

V	P	PC	PT = RR	Length
SSRC Receiver				
接收报告块 1				
⋮				
接收报告块 n				
$t^R req$				

(a)接收者(RTT_RR)报文

V	P	PC	PT = SR	Length
SSRC Sender				
SSRC Receiver1				
$t^R req1$				
$t^S req1$				
$t^S rep1$				
SSRC Receiver2				
$t^R req2$				
$t^S req2$				
$t^S rep2$				
…				

(b)发送者(RTT_SR)报文

图 1 改进后的 RTCP 报文格式
Fig.1 Improved message format of RTCP

RTT_RR 是由接收者发出的向发送者提出测试 RTT 的请求报文,包括接收者的标识 SSRC 和提出 RTT 测量的请求时间 $t^R req$。RTT_SR 为发送者的应答报文,包括发送者的标识 SSRC、各接收者的 SSRC 和其对应的请求时间 $t^R req$,以及接收者收到该请求报文的时间 $t^S req$ 和接收者发送对应应答报文的时间 $t^S rep$。

测量方法如图 2 所示。将时间分为轮,T_{RR}表示两次 RTT 应答(RR)的时间间隔也即每一轮。在新一次 RR 开始时,接收者根据上一次 RR 所提供的抑制参数确定一个随机时间 T_S,这个随机时间 T_S的选择我们将在下一小节中进行讨论。当一接收者在自己的定时器(定时值为 T_S)时间到之前发现已经有其它接收者向发送者发出 RTT 请求,该接收者就将定时器归零并取消自己即将发送的 RTT 请求。如果在定时器时间到前没有发现其它接收者发出 RTT 请求,该接收者就在 $t^R req$ 时刻向发送者发出 RTT 请求。发送者接收到这个 RTT 请求就记录下该请求到达的时间 $t^S req$ 和发送这个 RTT 请求的接收者的编号(SSRC)。当 T_{RR}时间到时,发送者向所有接收者多播 RTT 应答消息。RTT 应答消息包括该消息的发送时间 $t^S rep$、对应所有未抑制接收者的编号(SSRC)及对应的请求时间 $t^R req$ 和到达时间 $t^S req$、下轮 RR 的抑制参数(λ, T)。

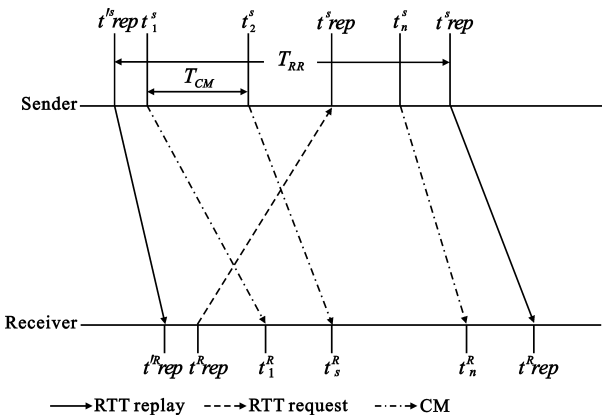

图 2 RTT 时序图
Fig.2 RTT time chart

在 T_{RR}时间间隔中,发送者会定时向接收者发送控制消息(CM)包,接收者可以根据包里的信息测量单向延时。其中,T_{CM}($T_{CM} << T_{RR}$)表示发送两个 CM 包的时间间隔;T_{CM}和 T_{RR}并不是固定不变的,发送者在每次 RR 中都会更新它们的值。

我们定义 CM 包的格式如图 3 所示,包括发送者的 SSRC 和 CM 包发出的时间。

V	P	PC	PT = CM	Length
SSRC Sender				
t^S_k				

图 3 控制信息(CM)报文格式
Fig.3 Message format of CM

在接收到发送者的 RTT 应答报文后,发送 RTT

请求报文的接收者就可以计算出闭环的 RTT 值:

$$t_{RTT} = d_{RS} + d_{SR} \tag{1}$$

式中,d_{RS}表示从接收者到发送者的单向延时,d_{SR}表示从发送者到接收者的单向延时。

d_{RS}和 d_{SR}可以定义如下:

$$d_{RS} = t^S req - t^R req + t_{off} \tag{2}$$

$$d_{SR} = t^R rep - t^S rep - t_{off} \tag{3}$$

$$d_{RS} = d_{SR} - t_{asy} \tag{4}$$

式中,t_{off}和t_{asy}分别表示时钟偏差和链路不对称性造成的时间延时。由上面的式子我们得到:

$$t_{RTT} = t^R rep - t^R req - (t^S rep - t^S req) \tag{5}$$

由此我们看出 RTT 的值与 t_{off}和 t_{asy}无关。

从式(2)~(4)我们可以得到时钟偏差和链路不对称性引起的延时值的和,该值用在下一小节单向延时测量中。

$$\omega = 2t_{off} + t_{asy} = t^R rep - t^S rep - t^S req + t^R req \tag{6}$$

t_{off}和t_{asy}的值并不是固定的,会随着时钟频差和动态网络发生突变而改变。

3　单向延时测量

当接收者接收到第 n 个 CM 包时,就可以估算其单向延时:

$$d'_{SR} = t_n^R - t_n^S - t'_{off} \tag{7}$$

根据上节中所讨论的 ω 值,我们可以用单向延时来估算 RTT 值。

$$\begin{aligned} t'_{RTT} &= d'_{SR} + d'_{RS} \\ &= d'_{SR} + (d'_{SR} - t'_{asy}) \\ &= 2(t_n^R - t_n^S - t'_{off}) - t'_{asy} \\ &= 2(t_n^R - t_n^S) - (2t'_{off} + t'_{asy}) \\ &= 2(t_n^R - t_n^S) - \omega' \end{aligned} \tag{8}$$

假设 $\omega' = \omega$,得到:

$$t'_{RTT} = 2(t_n^R - t_n^S) - \omega \tag{9}$$

同样,当接收到 RR 包时,被抑制的接收者也可以计算出单向延时,并用单向延时来估算 RTT 的值。d_{RS}的值在每个 T_{CM}中都会更新,而 t_{off}和 t_{asy}的值是在每一次 RTT 估算后才更新。由于时钟频差和动态网络发生突变的原因,最新更新的时钟偏差值和网络链路不对称引起的延时值会与当前值不吻合。

从上面的测量方法中我们可以看到,频率较小的进行 RTT 测量可以避免反馈内爆,而频率较大的进行单向延时测量可以确保 RTT 值更新的准确性。为了避免一次 RTT 时间内的测量样本对多播组发送产生过多的影响,我们使用指数加权滑动平均算法来平滑 RTT 时间的测量结果:

$$\bar{t}_{RTT} = (1 - \alpha)\bar{t}_{RTT} + \alpha t_{RTT} \tag{10}$$

在 RTT 测量中,我们设置 $\alpha = 0.5$;在单向延时测量中,取 $\alpha = 0.1$。

4　反馈时间 T_S的选定

在第 2 节中,我们提到了一个随机时间 T_S,T_S的选择是算法中一个很重要的组成部分。如果发出 RTT 请求的接收者数量比较多,那么就容易引起反馈内爆;如果数量偏少,那么就对比较准确测量 RTT 的值有一定影响。因此,选择合适的 T_S可以保证较准的 RTT 测量精度。

我们用一个右截的指数分布函数[5]来定义随机时间 T_S,其密度函数为

$$f(x) = \frac{1}{e^{\lambda} - 1} \cdot \frac{\lambda}{T} e^{\frac{\lambda}{T}x}, x \in [0, T] \tag{11}$$

通过式(11)得到的参数值,使得在我们的测量方法中10 000个接收者只能发出不到 10 个的反馈报文,数量很小的反馈报文也意味着很少的 RTT 测量次数。为了平衡反馈内爆和 RTT 测量准确性的关系,我们对 RTT 测量方法进行优化。

对于有着较小 RTT 估算值的接收者,由于其可以较早得到 RR 报文和更趋向于发送 RTT 请求报文,我们可以改变 T 和λ 的值或者增加 T_S的值。这里我们用增加 T_S的值来进行优化:$T_S = T'_S + (d_{\max} - d_{SR})$,其中 T'_S 为理论计算值,$d_{\max}$为上一轮测量中所有接收者计算出的单向延时的最大值,d_{SR}为上一次该接收者估算的单向延时。

对于那些被抑制而不能发出 RTT 请求的接收者,我们用优先级因子 p 保证其有较高的优先级来发送 RTT 请求。在加入多播会话后,加入的接收者维护自己的优先级因子 p,p 的值最开始赋值为一个最大值 $p_{\max}$。如果某接收者进行了一次 RTT 测量,那么 $p = 1$,否则在某轮 RR 结束后,其 p 的值改变为$p + 1$。如果在最新一次单向延时测量中,得到的测量值超过了一个设定好的极限值,这意味着网络发生了突变,那么接收者就要设置 $p = (1 + b)p$ $(0 < b < 1)$,提高优先级因子来更新 t_{off}和 t_{asy}的值。

$$p = \begin{cases} 1, & \text{往返} \\ p + 1, & \text{单向} \\ (1 + b)p, & \text{门限} \end{cases} \tag{12}$$

于是,优化改进后的 T_S值为

$$T'_S = (1 - q)(T_S + d_{\max} - d_{SR}) + \frac{q}{p}T, 0 < q < 1 \tag{13}$$

式中，q 表示 T 的偏差量。

5 仿真实验

我们对上述测试方法的有效性和对抑制的影响进行仿真实验。首先设置了 3 个多播会话，每个多播组的成员数目分别是 100、1 000和10 000。图 4 显示的是在每一轮 RR 中所发出的 RTT 请求的数量，可以看出，当组成员数目较小时，未被抑制的接收者占总接收者数量的9%以下，而当组成员个数增大时，这个比率随之降低，占总接收者数量的2%以下，这样不会引起反馈内爆，保证了网络不会形成拥塞。

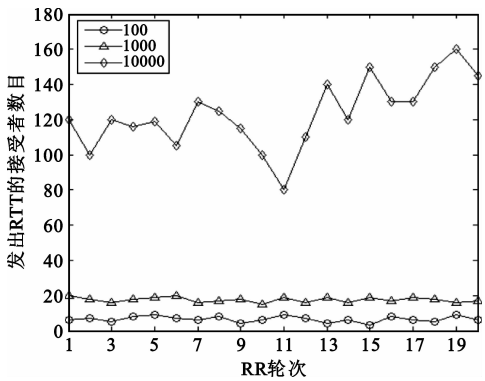

图 4 测量方法对抑制的影响
Fig.4 Suppression effect

为了估算本文方法的准确性，我们建立了一个多播会话，组成员数目为 10。接收者分别用本文方法和普通的 RTT 测量方法对 RTT 进行估算。多播连接时间持续200 s，每隔10 s就对本文 RTT 测量方法计算得到的 RTT 值 t' 和用普通测量方法得到的 RTT 值 t 进行记录。图 5 显示的是其相对偏差 $|t-t'|/t$，从图中我们可以看到其值不超过 10%，在可以接收的范围内，因此本文提出的算法是可行的。

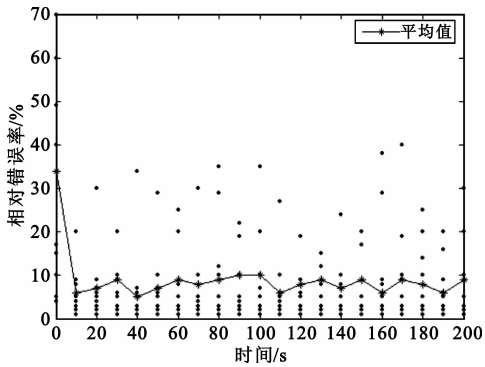

图 5 测量方法准确性
Fig.5 Accuracy of measurement method

6 结 论

RTT 的计算对于多播拥塞控制方案如何保证多播会话与 TCP 会话共存时的公平性有很重要的作用，这也是分层多播方案能在互联网上广泛运用的关键所在。本文提出了一种新的 RTT 计算方法，该算法将 RTT 测量与单向延时测量结合起来一起估算 RTT 值，仿真表明这种方法是可行的。在这种 RTT 测量方法中，反馈时间 T_S 由右截的指数分布函数来选定，有效地避免了反馈内爆，但却降低了 RTT 测量的准确性，下一步工作中，如何选择 T_S，有效平衡反馈内爆和 RTT 测量准确性的关系，将是我们研究的重点。

参考文献：

[1] Jamshid Mahdavi, Sally Floyd. TCP - friendly unicast rate - based flow control[EB/OL]. 1997 - 01 - 08. http://www.psc.edu/networking/papers/tcp_friendly.html.

[2] Padhye J, Firoiu V, Tow sley D, et al. Modeling TCP throughput: a simple model and its empirical validation[C]// Proceedings of ACM SIGCOMM′98. Vancouver, Canada: ACM, 1998: 303 - 315.

[3] RFC 1889, RTP: A Transport Protocol for Real - Time Applications[S].

[4] Liu J, Li B, Zhang Y. A Hybrid Adaptation Protocol for TCP - friendly Layered Multicast and Its Optimal Rate Allocation [C]//Proceedings of IEEE INFOCOM′02. New York, USA: IEEE, 2002: 1520 - 1529.

[5] Nonnenmacher J, Biersack E W. Scalable feedback for large groups[J]. IEEE/ACM Transactions on networking, 1999, 7 (3): 375 - 386.

作者简介：

张 牧(1980 -)，男，贵州贵阳人，2007 年于南京邮电大学获博士学位，现为副教授，主要研究方向为计算机网络；

ZHANG Mu was born in Guiyang, Guizhou Province, in 1980. He received the Ph.D. degree from Nanjing University of Posts and Telecommunications(NUPT) in 2007. He is now an associate professor. His research direction is computer network.

Email: zhangmu8045@126.com

王 攀(1979 -)，男，河南郑州人，南京邮电大学博士研究生，主要研究方向为计算机网络。

WANG Pan was born in Zhengzhou, Henan Province, in 1979. He is currently working toward the Ph.D. degree at NUPT. His research direction is computer network.