

文章编号:1001-9081(2011)12-3195-05

doi:10.3724/SP.J.1087.2011.03195

基于种群生态的组播拥塞控制机制

麻海圆^{1,2}, 孟相如¹, 马志强¹, 周晶³, 庄绪春¹

(1. 空军工程大学 电讯工程学院, 西安 710077; 2. 94303 部队, 山东 潍坊 261051;

3. 北京邮电大学 信息与通信工程学院, 北京 100876)

(haiyuanma@live.cn)

摘要:针对现有TCP类组播拥塞控制机制不具有速率平滑性、往返时间(RTT)公平性以及在高速环境中传输效率低的问题,提出一种基于种群生态理论的自适应高速组播拥塞控制机制。该机制在每个接收端实现瓶颈链路带宽和背景流速率的测量,并将这两个测量值用于种群生态模型中以计算期望服务速率,然后使用一种简单的反馈抑制机制选取期望服务速率最小的接收端作为代表,该代表将其期望服务速率反馈给源端控制发送速率。仿真结果表明新机制发送速率平滑,具有RTT公平性,在低速网络和高速网络中都能与单播流公平共享带宽资源。

关键词:组播;拥塞控制;种群生态;期望服务速率

中图分类号: TP393.03 **文献标志码:**A

High-speed multicast congestion control based on population ecology

MA Hai-yuan^{1,2}, MENG Xiang-ru¹, MA Zhi-qiang¹, ZHOU Jing³, ZHUANG Xu-chun¹

(1. Institute of Telecommunication Engineering, Air Force Engineering University, Xi'an Shaanxi 710077, China;

2. Troops 94303, Weifang Shandong 261051, China;

3. School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: The existing TCP multicast congestion control mechanisms do not have speed smoothness, Round-Trip Time (RTT)-fairness, and high efficiency in high-speed networks. To solve these inherent disadvantages derived from TCP, an Adaptive High-Speed Multicast (AHSM) congestion control mechanism was proposed based on population ecology theory. The new mechanism estimated the bottleneck bandwidth and the rate of background traffic at each receiver. Then the estimated values were used in population ecology model to calculate requested service rate. The receiver with lowest requested service rate was selected as representative to control the sending rate via feedbacks. The simulation results show that the new mechanism has speed smoothness and RTT-fairness, it also has high transmission efficiency in high-speed networks. It can share bandwidth with unicast flow in both low and high-speed networks.

Key words: multicast; congestion control; population ecology; requested service rate

0 引言

IP组播具有延迟小、冗余流量少的优点,是承载多点业务的最佳机制。组播承载的业务大多是基于流媒体的音/视频应用,如网络电视、视频会议、远程教学、数字影院等。与一般的数据业务不同,流媒体业务对服务质量(Quality of Service, QoS)有特殊的需求。首先,其编码和压缩技术要求平滑的传输速率,否则会加重终端解码的负担,并导致接收质量下降;其次,其业务数据量大,要求较高的传输速率。

组播拥塞控制机制是保证组播QoS的关键技术之一,也是维护网络稳定、避免拥塞、充分发挥网络效能的重要保证。为了保证与TCP流的友好性,现有的组播拥塞控制机制多数是模拟TCP的和式增加积式减少(Additive Increase Multiplicative Decrease, AIMD)机制或使用TCP吞吐量公式^[1],本文将此类组播机制称为TCP类机制^[2]。这些机制具有TCP的共同缺陷,即速率抖动和不具有往返时间(Round-Trip Time, RTT)公平性,随着网络不断朝着高速化的方向发展,这些机制传输效率低的问题也愈发突出。为此,本文借鉴

种群生态模型的自适应性,提出一种自适应高速组播拥塞控制机制(Adaptive High-Speed Multicast congestion control, AHSM)。

1 相关研究

组播拥塞控制可以分为单速率机制和多速率机制。由于大多数多速率机制是由单速率机制扩展而来,且单速率机制操作简单,易于实现,因此单速率机制仍然是研究热点。近年来对单速率组播拥塞控制的研究重点在于提高速率的平滑性,其中大多数是对TFMCC(TCP-Friendly Multicast Congestion Control)^[3]的改进。如文献[6]使用加权滑动平均对TFMCC的发送速率进行平滑,并依据时延抖动判断拥塞状态,然后调整加权系数;文献[7]使用马尔可夫过程对AIMD机制进行建模,得出一个新的TCP吞吐量公式,将此公式代替TFMCC中使用的TCP吞吐量公式,并依据RTT的变化进一步平滑发送速率;文献[8]依据路由器和接收端的反馈,使用模糊控制器对TFMCC的发送速率进行平滑。这些机制在一定程度上改善了组播的速率平滑性,但是都没能解决TCP类组播机制的

收稿日期:2011-06-17;修回日期:2011-08-16。

基金项目:国家科技重大专项(2009ZX03003-006-04);陕西省自然科学基金资助项目(SJ08F14;2009JQ8008)。

作者简介:麻海圆(1982-),男,河南南阳人,博士研究生,主要研究方向:宽带通信网络、拥塞控制;孟相如(1963-),男,陕西蓝田人,教授,博士生导师,主要研究方向:宽带通信网络、网络故障诊断;马志强(1968-),男,山东烟台人,副教授,主要研究方向:视频通信;周晶(1983-),男,山东潍坊人,博士研究生,主要研究方向:移动通信;庄绪春(1980-),山东曲阜人,讲师,硕士,主要研究方向:宽带网络。

固有不足:一是在高速网络中传输效率低;二是不具有 RTT 公平性。TFMCC 中使用的 TCP 吞吐量公式是对 AIMD 的建模,因此具有 AIMD 的共性,当具有不同 RTT 的多个流竞争同一瓶颈链路时,具有较长 RTT 的流将难以获得公平的服务速率。另外,为了抑制反馈,并不是所有接收端都能向源端发送反馈,因此这些接收端的 RTT 测量也是困难的。

为了彻底消除 RTT 不公平性问题,并提高高速网络中组播的传输效率,应当摒弃模拟 TCP 的设计思想,提出全新的组播拥塞控制机制。文献[11~12]使用种群生态模型设计单播传输控制机制,为本文提供了借鉴。种群生态学是研究生物群落与环境自然,以及种群之间相互关系的理论。在有限的环境资源下,各种群无需外界的干预自适应地实现资源的公平共享。比较成熟的种群生态模型有 Logistic 模型和 Lokta-Volterra (LV) 模型。

Logistic 模型为

$$\frac{dx(t)}{dt} = \gamma x(t)(1 - \frac{x(t)}{K}) \quad (1)$$

式中: $x(t)$ 是种群规模, γ 为自然增长率, K 为种群规模的上限。Logistic 模型认为种群规模受环境资源的限制,其增长率 $\frac{dx(t)}{dt}$ 随着种群规模的增大而减小,当达到上限 K 时,增长率 0。若种群规模超过 K , 种群将会减小。

Logistic 模型仅模拟了单种群的增长规律。LV 模型对多种群的相互关系进行建模,更符合实际。两个相互竞争种群的 LV 模型为

$$\begin{cases} \frac{dx_1(t)}{dt} = \gamma_1 x_1(t) \left(1 - \frac{x_1(t) + \omega_{1,2}x_2(t)}{K}\right) \\ \frac{dx_2(t)}{dt} = \gamma_2 x_2(t) \left(1 - \frac{x_2(t) + \omega_{2,1}x_1(t)}{K}\right) \end{cases} \quad (2)$$

式中: γ_i 是第 i 个种群的自然增长率; $\omega_{i,j}$ 是第 j 个种群相对于第 i 个种群的竞争因子,反映了两个种群之间的竞争关系。可见 Logistic 模型是 LV 模型在单种群时的特例。从式(2)中可以看出,若 $\omega_{i,j} = 1$, 则当两个种群规模之和超过 K 时, 种群开始减小。这与多个流竞争带宽资源的情形完全一致。

文献[11]将流的发送速率模拟为种群规模,将所有的背景流视为一个竞争种群,将瓶颈链路带宽 K 模拟为种群上限,直接使用 LV 模型控制发送速率。在文献[12]中,每一个路由器将自己的聚合流速率和队列长度计算成一个虚负载因子,并将该值记录在包头中传送给接收端,接收端再将其反馈给源端,源端将此值用于变异的 LV 模型调节发送窗口。这两种机制在速率平滑性、公平性和稳定性上都表现出了优秀的性能。但是这两种机制都难以直接扩展到组播中。文献[11]假设瓶颈链路容量是已知的,在实际中,若没有链路容量测量工具,终端无法获知瓶颈链路带宽;而且源端需要依据接收端的反馈信息探测可用带宽,在组播中不允许所有接收端都发送反馈,这种探测方法无法应用。文献[12]所提协议需要对所有路由器进行改造,因此难以部署。

本文借鉴这两篇文献的思想,首先根据组播的特点,在每个接收端实现瓶颈链路带宽和背景流量的测量,然后将这两个值用于种群生态模型以控制源端发送速率。

2 AHSM 机制

2.1 系统控制流程

AHSM 是端到端的控制机制,无需中间节点的支持,所有

控制功能均在端系统实现,尤其是大部分的操作都在接收端实现。

如图 1 所示,源端在收到代表(Representative)的反馈包后,直接将发送速率 R_s 调节为代表的期望服务速率 R_{req}^R 。如果反馈包是非代表发送的,并且其期望服务速率满足 $R_{req} < \beta R_s$, 则更换代表。 β 是抑制系数,其作用将在 2.5 节中说明。源端的另一个工作是发送探测包对,供接收端探测瓶颈链路带宽。

接收端依据探测包对的接收时间间隔测量瓶颈链路带宽 C , 依据数据包的时延偏差计算背景流量 $b(t)$, 然后使用种群生态模型计算期望服务速率 R_{req} 。由于 R_{req} 隐含了瓶颈链路带宽和背景流速率,因此反映了接收端实际接收能力。为了保证 RTT 公平性,接收端以固定的时间周期 τ 计算 R_{req} , 而不是以 RTT 为周期。每个周期结束时,依据反馈抑制机制决定是否发送反馈,如果可以发送反馈,则将自己的期望服务速率封装在反馈包中发送给源端。

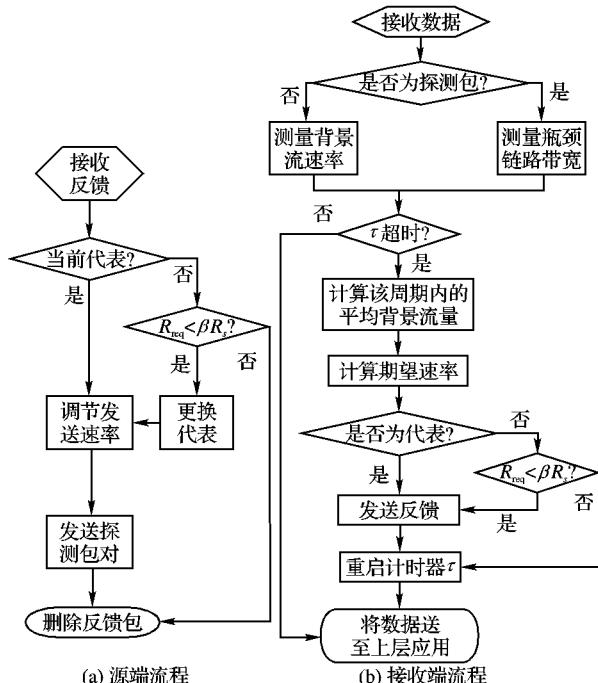


图 1 AHSM 控制流程

2.2 基于种群生态的速率控制模型

设 t 时刻组播的发送速率为 $x(t)$, 某接收端瓶颈链路上的背景流速率为 $b(t)$, 则瓶颈链路的聚合流速率为 $a(t) = x(t) + b(t)$ 。如果将这三者分别模拟为种群规模、竞争种群规模和种群上限,可将 LV 模型转化为

$$\frac{dx(t)}{dt} = \gamma x(t) \left(1 - \frac{x(t) + \omega b(t)}{C}\right) \quad (3)$$

其差分模型为

$$x(t) = x(t - \tau) + \gamma x(t - \tau) \left[1 - \frac{x(t - \tau) + \omega b(t - \tau)}{C}\right] \quad (4)$$

在此模型中,瓶颈链路带宽 C 和背景流速率 $b(t)$ 需要通过测量工具得到,这也是 AHSM 要解决的重点。下面分别提出对这两个变量的测量方法,由于组播中大部分接收端不能发送反馈,因此测量方法都在接收端实现,仅需由源端到接收端的单向信息流动,即单程测量方式。

2.3 瓶颈链路带宽测量

文献[13]提出一种简单快速测量瓶颈链路带宽的方法。

源端发送一对背靠背(即连续的)的探测包,由于两个数据包是连续发送的,包之间没有间隔,因此在传输过程中,两个包被其他流分开的可能性很小,因此它们到达接收端的时间间隔 μ 就是瓶颈链路服务一个数据包所需要的时间。设包的长度为 S ,则瓶颈链路的链路带宽即为

$$C = S/\mu \quad (5)$$

该方法在接收端实现,无需反馈信息,因此非常适合于组播。其突出优势是测量速度快,理论上仅需要一组探测包对就能得到链路容量。本文将使用该方法测量链路容量,并将其用于 LV 模型中。

当组规模很大时,探测包会在网络中产生大量的额外流量。因此,AHSM 不使用专门的探测包对,而是通过调整原有数据包的发送方式实现。AHSM 在数据包头中设置一个标记位 packet_type,用 0 表示正常数据包,1 表示第一个探测包,2 表示第二个探测包。在每个速率调节周期,源端首先按照背靠背的方式发送一对数据包,分别标记为 1 和 2,然后按照代表反馈的速率发送数据包,均标记为 0。接收端收到标记有 1 和 2 的包后即可测量瓶颈链路带宽。

虽然文献[13]认为探测包对被背景流分开的可能性很小,但是不能完全排除这种可能性。一旦两个探测包之间插入了其他包,终端得到的 μ 将大于瓶颈链路服务一个包所需要的时间。为了避免这种情况对测量瓶颈链路容量带来偏差,AHSM 记录最近得到的 M 个服务时间 $[\mu_k, \mu_{k-1}, \dots, \mu_{k-M+1}]$,在使用式(5) 测量瓶颈带宽时,选取其最小值,即 $\mu = \min_i \mu_i (i = k - M + 1, k)$ 。这样,即使偶尔探测包对被背景流分开,也不会影响测量瓶颈链路带宽的准确性。

2.4 背景流速率测量

LV 模型中另一个重要的变量是瓶颈链路上的背景流速率。本文提出一种利用数据包的单向时延测量背景流速率的方法。该方法在接收端实现,且无需任何探测包。

若时间间隔 τ 很短,可以认为在 $[t - \tau, t]$ 内 $x(t)$ 和 $b(t)$ 为定值。假设瓶颈链路队列始终不为空,即排队时延 $q(t)$ 恒大于 0,则队列长度的变化量与聚合流速率的关系为:

$$q(t) - q(t - \tau) = [a(t - \tau) - C]\tau \quad (6)$$

设 t 时刻接收端收到第 i 个包,该包的发送时间和接收时间分别为 $T_s(i)$ 和 $T_r(i)$ 。用该包的单向时延表示 t 时刻的单向时延,即 $d_{owd}(t) = d_{owd}(i)$;同理, $t - \tau$ 时刻接收到第 j 个包($j < i$),用该包的单向时延表示 $t - \tau$ 时刻的单向时延,即 $d_{owd}(t - \tau) = d_{owd}(j)$ 。定义 t 时刻的单向时延差为:

$$\begin{aligned} e(t) &= d_{owd}(t) - d_{owd}(t - \tau) = d_{owd}(i) - d_{owd}(j) = \\ &= [T_r(i) - T_s(i)] - [T_r(j) - T_s(j)] = \\ &= [T_r(i) - T_r(j)] - [T_s(i) - T_s(j)] \end{aligned} \quad (7)$$

单向时延由传输时延 d_{trans} 、处理时延 d_{proc} 、时钟偏差 d_{skew} 和排队时延 d_{queue} 组成。如果端到端的传输路径恒定不变,则前三项是定值,即

$$d_{owd}(t) = d_{trans} + d_{proc} + d_{skew} + d_{queue}(t) \quad (8)$$

因此

$$\begin{aligned} e(t) &= d_{owd}(t) - d_{owd}(t - \tau) = d_{queue}(t) - d_{queue}(t - \tau) = \\ &= [q(t) - q(t - \tau)]/C \end{aligned} \quad (9)$$

即单向时延差等于排队时延之差。

结合式(6)和式(9)可得聚合流速率为

$$a(t - \tau) = C \left(1 + \frac{e(t)}{\tau} \right) \quad (10)$$

则背景流速率为

$$\begin{aligned} b(t - \tau) &= a(t - \tau) - x(t - \tau) = \\ &= C \left(1 + \frac{e(t)}{\tau} \right) - x(t - \tau) \end{aligned} \quad (11)$$

因此,通过观察单向时延差就可以测量背景流速率。由式(7)可以看出,计算时延偏差不需要源端和接收端的同步,且与传输时延 d_{trans} 无关,因此也与 RTT 无关,这就保证了 AHSM 的速率调节与 RTT 无关。

在得到瓶颈链路带宽和背景流速率后就可以使用种群生态模型计算期望服务速率了。然而,2.3 节中测量背景流速率的方法建立在队列始终不为空的假设下。若队列为空则这种估计方法将会失效。在 AHSM 中,如果时间段 τ 内的排队时延为 0,即该时间段内瓶颈链路队列始终为空,说明聚合流速率远小于链路容量,某一个流速率的快速增加不会对其他流造成竞争,此时使用单种群的 Logistic 模型更新速率。将式(1)改为差分模型得:

$$x(t) = x(t - \tau) + \gamma x(t - \tau) \left(1 - \frac{x(t - \tau)}{C} \right) \quad (12)$$

综上,速率控制方程为:

$$\begin{cases} x(t) = x(t - \tau) + \\ \gamma x(t - \tau) \left(1 - \frac{x(t - \tau)}{C} \right), & d_{queue}(t) = 0 \\ \gamma x(t - \tau) \left[1 - \frac{x(t - \tau) + \omega b(t - \tau)}{C} \right], & d_{queue}(t) > 0 \end{cases} \quad (13)$$

接收端将式(13)计算得到的速率值 $x(t)$ 作为自己的期望服务速率 R_{req} 。

2.5 反馈抑制

当组规模很大时,必须使用有效的反馈抑制机制避免反馈内爆。端到端的单速率机制大都采用代表控制源端发送速率,即选择全组中期望服务速率最小的接收端作为代表,该代表定期将自己的期望服务速率反馈给源端。AHSM 同样也使用代表机制。源端将当前的发送速率 $R_s(t)$ (即当前代表的期望速率)和当前代表的地址封装在包头中组播给所有接收端。作为代表的接收端会以固定的周期 τ 发送反馈。而非代表通过判断自己的期望速率是否小于当前的发送速率决定是否发送反馈。为了避免动态网络环境下代表在几个期望服务速率接近的接收端之间切换,AHSM 设置一个略小于 1 的抑制系数 β ,只有当

$$R_{req} < \beta R_s(t) \quad (14)$$

时才会发送反馈。同样,当源端接收到一个非代表的反馈时,只有当其期望速率满足式(14)时才将此接收端作为新的代表,否则将忽略该反馈。这样就避免了代表的频繁切换,同时也减少了反馈包。

3 性能分析

本节通过 NS2^[14]仿真工具分析 AHSM 的性能,并将其与 TFMCC 进行比较。重点讨论新机制的速率平滑性、RTT 公平性和高速传输特性。仿真使用哑铃式单瓶颈网络拓扑,如图 2 所示。图中包含 4 个单播源端和一个组播源端,四个接收端分别与四个单播源端建立单播连接,同时加入组播会话。仿真中包的大小均为 1 000 字节,瓶颈链路采用 DropTail 队列类型。AHSM 的主要参数为: $M = 50, \beta = 0.95, \tau = 50 \text{ ms}, \gamma = 1.95^{[11]}, \omega = 0.9^{[11]}$ 。

3.1 速率平滑性和 TCP 友好性

首先考查 AHSM 发送速率的平滑性,使用 TCP 作为竞争

流，并观察 TCP 友好性。图中两个路由器之间的链路带宽为 5 Mbps，其余所有链路的带宽均为 100 Mbps，所有链路时延为 10 ms。因此组播流和每个单播流的公平速率为 1 Mbps。在仿真开始时建立四条 TCP 连接，100 s 是建立组播会话。图 3 显示的是分别使用两种协议时组播吞吐量与其中一个 TCP 吞吐量。从图 3 中可以看出，在低速链路中，AHSM 和 TFMCC 都具有 TCP 友好性，各自吞吐量都在公平速率附近抖动。与 TFMCC 相比，AHSM 速率非常平滑，速率标准差仅为 0.085，而 TFMCC 速率的标准差达到 0.388，是 AHSM 的 4.5 倍。因此，AHSM 非常适合传输流媒体业务。

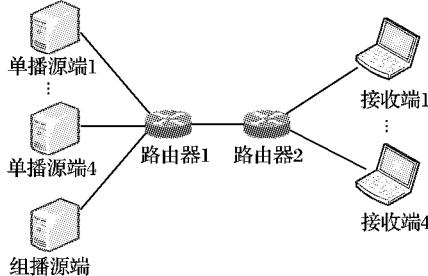


图 2 单瓶颈网络拓扑

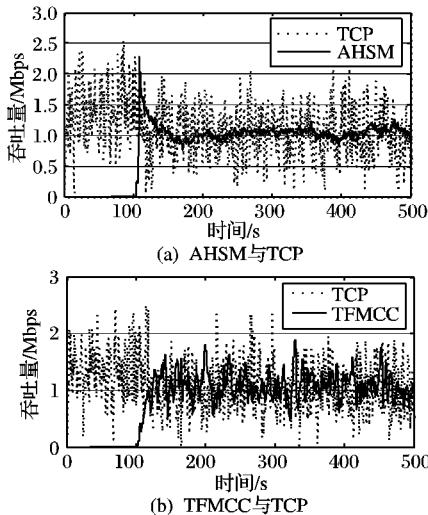


图 3 速率平滑性的比较

3.2 RTT 公平性

本实验考查 AHSM 的 RTT 公平性。在图 2 的单瓶颈网络中，瓶颈链路带宽为 4 Mbps，路由器 2 与四个接收端之间的传输时延分别为 5 ms、50 ms、100 ms 和 200 ms，组播源端与路由器 1 以及两个路由器之间的链路时延均为 5 ms。为了使四个接收端具有相同的可用带宽，该实验中不在终端之间建立 TCP 连接，而是在两个路由器之间使用 100 个 on/off 流量模型产生平均为 2 Mbps 的背景流，用以模拟 Web 流。从而，四个组播接收端的可用带宽完全相等，都为 2 Mbps，但是 RTT 不同，最小 RTT 分别为 30 ms、120 ms、220 ms 和 420 ms。图 4 比较了分别使用两种协议时组播的吞吐量以及四个接收端计算的期望服务速率。从图 4(a)中可以看出，用 AHSM 时，四个接收端的期望服务速率与组播的吞吐量几乎完全一致（五条曲线重合），吞吐量的平均值为 2.04 Mbps。而 TFMCC 使用 TCP 吞吐量公式计算期望速率，四个组成员经历相同的丢率，但具有不同 RTT，因此期望速率也各不相同。尤其是具有较短 RTT 的接收端的期望速率已经远大于实际的可用带宽（图 4(b)中的 T_1 ）。而具有较长 RTT 的接收端的期望速率又小于可用带宽（图 4(b)中的 T_4 ），致使 TFMCC 的平均吞吐

量仅为 1.63 Mbps，比实际可用带宽小 18.5%。

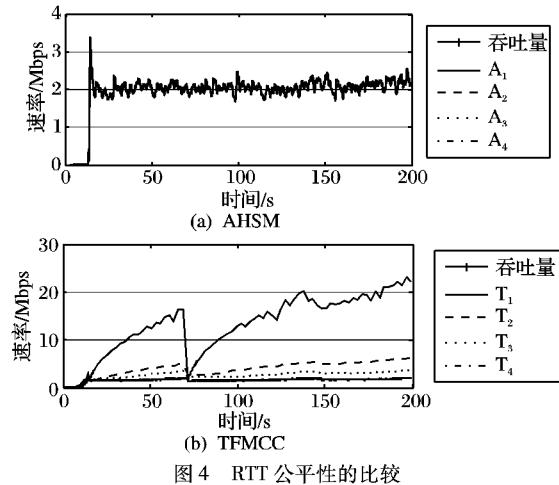


图 4 RTT 公平性的比较

AHSM 具有非常优良的 RTT 公平性，这得益于其速率调节周期及调节量均与 RTT 无关，也得益于其能准确探测得到瓶颈链路带宽和背景流速率。仿真表明，四个接收端均能十分精确地测量出瓶颈链路带宽为 4 Mbps。图 5 显示的是具有最短 RTT 的流 (A_1) 和最长 RTT 的流 (A_4) 在接收端测量出来的背景流速率。可见，两者之间十分接近，仅存在时延上的偏差。通过统计，四个接收端测量的背景流速率的平均值分别为 1.9407 Mbps、1.9414 Mbps、1.9415 Mbps 和 1.9416 Mbps，与实际背景流之间的误差不超过 3%。说明只要各接收端经历相同的瓶颈链路，无论 RTT 是多少，其测量得到的背景流速率都是相同的。

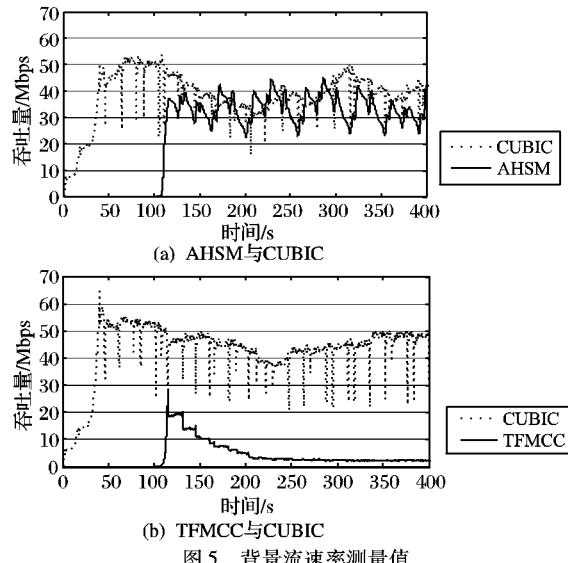


图 5 背景流速率测量值

3.3 高速传输性能

AHSM 的另一个突出性能是在高速网络中具有较高的传输效率。当前 Linux 系统中应用最广泛的高速单播传输协议是 CUBIC^[15]，因此在仿真中使用 CUBIC 作为竞争流分析协议性能（在高速网络中讨论与 TCP 的竞争是毫无意义的）。在图 2 的单瓶颈网络中，两个路由器之间的链路带宽为 200 Mbps，其余所有链路带宽为 1000 Mbps，所有链路时延均为 50 ms。因此组播和每条 CUBIC 流的公平带宽为 40 Mbps。四条 CUBIC 流在仿真开始时建立连接，组播流在 100 s 建立。图 5 比较了分别使用 AHSM 和 TFMCC 时组播吞吐量与其中一条 CUBIC 流的吞吐量。可以看出，AHSM 的吞吐量在 117 s 时达到 36 Mbps，之后基本与 CUBIC 持平。AHSM 平均吞吐

量为 34.06 Mbps, 四个 CUBIC 流的平均吞吐量为 39.89 Mbps, 略高于 AHSM。因此 AHSM 具有较高的传输效率和较好的友好性。而 TFMCC 在高速网络中仅获得平均 2.59 Mbps 的吞吐量, 这是由于 TFMCC 使用 TCP 吞吐量公式调节发送速率, 无法满足高速网络中的传输需求, 这是 TCP 及 TCP 类组播机制的固有缺陷。

从本实验和 3.1 节可以看出, AHSM 能够在低速环境中与 TCP 公平共存, 在高速环境中与 CUBIC 公平共存。需要指出的是, 近年来提出的一些高速单播协议, 如 High Speed TCP^[16]、Binary increase TCP^[17], 为了实现在两种环境下都具有公平性, 需要设置一个拥塞窗口阈值在高速模式和低速模式之间切换。而 AHSM 在两种环境中的公平性完全是自适应的, 不需要修改参数, 也无需模式切换, 这归功于其能准确测量瓶颈链路带宽和背景流速率。

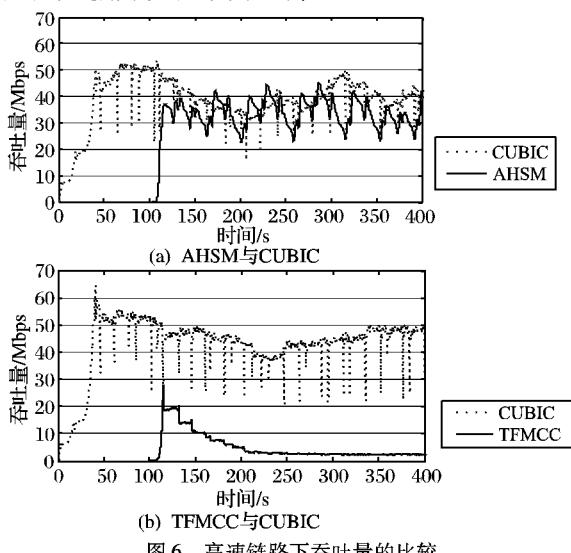


图 6 高速链路下吞吐量的比较

3.4 计算复杂度分析

从图 1(a) 中可以看出, 本协议源端操作非常简单, 仅仅是依据代表的反馈调节发送速率。因此, 影响源端操作复杂度的主要因素是反馈数。实验证明, 本协议的反馈抑制效果明显优于 TFMCC。在图 2 所示的网络拓扑中, 如果组规模由 4 个增长到 128, 则 AHSM 每秒钟收到的平均反馈数由 20.6 增长至 48.3, 仅增长了 1.3 倍; 而当组规模为 4 时, TFMCC 的反馈为平均每秒 42.4, 是 AHSM 的 2.1 倍; 当组规模达到 128 时, TFMCC 的平均反馈增加到 377.6 个, 是 AHSM 的 7.8 倍。可见 AHSM 减少了大量的反馈, 必然在很大程度上减轻源端负担, 减少网络冗余流量。

AHSM 接收端在测量瓶颈链路带宽时, 需要记录最近 M 个测量值并取最小值, 并对数组进行移位和比较操作, 其计算复杂度为 $O(M)$ 。但是这些操作在一个周期内仅执行一次, 而不是每收到一个包就执行一次, 因此计算量也很小; 在测量背景流速率时, 每成功接收一个包, 则依据式(11)计算一次背景流, 在每个周期结束计算本周期内的平均背景流速率, 然后使用式(13)调节发送速率, 这些操作都是简单的线性计算, 计算复杂度均为 $O(1)$ 。

4 结语

组播拥塞控制机制是当前业界的研究热点问题, 而对高速网络下组播拥塞控制的研究却相对较少。本文基于种群生态理论提出一种新的单速率组播拥塞控制机制。针对组播的

特点, 实现了在接收端实现瓶颈链路带宽和背景流速率的测量, 并将其用于种群生态模型控制源端发送速率。仿真表明, AHSM 计算量小, 具有良好的 RTT 公平性、平滑的发送速率, 在低速网络和高速网络中都能与单播流公平地共享网络资源。AHSM 的主要功能在接收端实现, 因此易于扩展为接收端驱动的动态分层组播机制, 这将是下一步的研究重点。

参考文献:

- [1] PADHYE J, FIROIU V, TOWSLEY D F. Modeling TCP Reno performance: a simple model and its empirical validation [J]. IEEE/ACM Transactions on Networking, 2000, 8(2): 133–45.
- [2] RIZZO L. Pgmc: a TCP-friendly single-rate multicast congestion control scheme [C]// Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. New York: ACM, 2000: 17–28.
- [3] WIDMER J, HANDLEY M. Extending equation-based congestion control to multicast applications [C]// Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. New York: ACM, 2001: 275–285.
- [4] 刘郁恒, 陈广文, 胡严, 等. 一种在接收端实现的 TCP-Friendly 拥塞控制机制 [J]. 电子学报, 2005, 33(5): 835–841.
- [5] CHEN H L, LIU Z X, CHEN Z Q, et al. Extending TCP congestion control to multicast [J]. Computer Networks, 2007, 51(11): 3090–3109.
- [6] BOURAS C, GKAMAS A, KIOUMOURTZIS G. Adaptive smooth multicast protocol for multimedia transmission: implementation details and performance evaluation [J]. International Journal of Communication System, 2010, 23(3): 299–333.
- [7] KAMMOUN W, YOUSSEF H. Equation-based end-to-end single-rate multicast congestion control [J]. Annals of Telecommunications, 2010, 65(3): 219–231.
- [8] 李欢, 孟相如, 郑博, 等. 一种模糊控制的组播速率调节机制 [J]. 计算机工程与应用, 2010, 46(32): 97–99.
- [9] YAMAMOTO Y, ITSUMI H, YAMAMOTO M. Proxy-based multicast congestion control for Ad Hoc networks [C]// The 8th Asia-Pacific Symposium on Information and Telecommunication Technologies. New York: IEEE, 2010: 1–6.
- [10] 康巧燕, 王建峰, 王蓉, 等. 一种基于 Fuzzy-PID 控制的组播拥塞控制机制 [J]. 系统工程与电子技术, 2011, 33(2): 416–422.
- [11] IGUCHI T, HASEGAWA G, MURATA M. A new congestion control mechanism of TCP with inline network measurement [EB/OL]. [2011-02-20]. <http://www.anarg.jp/achievements/web2004/papers/t-iguti05icoin-CongestionControlInline.pdf>.
- [12] HUANG X M, LIN C, REN F Y. A novel high speed transport protocol based on explicit virtual load feedback [J]. Computer Networks, 2007, 51(7): 1800–1814.
- [13] CONG L, LU G, CHEN Y, et al. Queueing-based TCP congestion estimator [J]. IET Communications, 2010, 4(16): 1974–1986.
- [14] NS2: Network Simulator (Version: 2.26) [EB/OL]. [2011-03-20]. <http://www.isi.edu/nsnam>.
- [15] RHEE I, XU L. CUBIC: a new TCP-friendly high-speed TCP variant [J]. SIGOPS Operating Systems Review, 2008, 42(5): 64–74.
- [16] RFC 3649, High speed TCP for large congestion windows [S], 2003.
- [17] XU L, HARFOUSH K, RHEE K. Binary increase congestion control for fast, long distance networks [C]// The 23rd Annual IEEE International Conference on Computer Communications. New York: IEEE, 2004: 2514–2524.