

一种 REM 算法辅助的分层组播流量控制方案

董春玲¹, 朱晓丽²

(1. 山东师范大学人口资源与环境学院, 济南 250014; 2. 山东广播电视大学计算机与通信学院, 济南 250014)

摘要: 基于分层组播中公平速率分配算法实施过程中存在的问题以及分层组播协议策略中同步点的优化问题, 提出了将主动队列管理算法 REM 作为对端系统的辅助加入到分层组播流量控制中, 将分层组播同步点策略、满足 Max-Min 公平性要求的速率分配算法以及基于 REM 的显式拥塞指示技术有机地结合起来, 设计了一种基于速率的、由接收者和发送者混合驱动的分层组播流量控制方案。仿真实验结果表明该方案使得分层多速率组播在保证会话内、会话间公平性的前提下, 提高了流量控制机制的高效性和对网络状态适应的灵敏性。

关键词: 分层组播; Max-Min 公平性; REM; 流量控制

A REM-assisted Scheme for Layered Multicast Traffic Control

DONG Chunling¹, ZHU Xiaoli²

(1. College of Population Resources and Environment, Shandong Normal University, Jinan 250014;

2. College of Computer and Communication, Shandong Radio and TV University, Jinan 250014)

【Abstract】 Based on the problems of the implementation for Max-Min fair rate allocation algorithm in layered multicast and the problems of synchronization points optimization in layered multicast protocol strategy, a REM-assisted scheme of traffic control is devised. The synchronization points strategy of layered multicast, the rate allocation algorithm satisfying Max-Min fairness and the explicit congestion notification technique based on REM are combined together in this scheme. The results of simulation experiments show that this scheme can improve the effectiveness of layered multicast's traffic control mechanism and the sensitivity to the changes of the network condition with the precondition of intra-session fairness and inter-session fairness guaranteed.

【Key words】 Layered multicast; Max-Min fairness; REM; Traffic control

分层多速率组播协议多采用接收者驱动的速率和流量控制策略, 这种方式虽然有效地解决了异构性问题, 但是要通过接收者不断地加入和退出组播组来适应网络状态的变化, 导致组播树结构经常变化, 加重了路由和接收者的负担, 而且收敛速度慢, 因此本文设计了一种由发送端和接收端混合驱动流量控制模式, 在保证有效解决 TCP 和分层组播共存的异构网络公平性问题的前提下, 一方面在接收端通过加入或离开组播组来调整接收速率, 将各接收者的丢包率等状态信息形成反馈报文报告给发送方; 另一方面在发送端则根据接收端的反馈信息来计算并调整每个组播信道的发送速率。

1 需要解决的两个问题

1.1 Max-Min 公平速率分配算法实施中的问题

笔者曾在文献[2]中设计了一种适用于分层多速率组播与单播并存的异构网络环境的、分布式 Max-Min 公平速率分配算法, 本文拟将其嵌入进来作为获取公平性流量分配的主要手段。该算法的设计目标是通过统一的、满足 Max-Min 公平性要求的速率分配和流量控制来保证共享同一瓶颈链路的各会话能公平分享链路带宽。然而该算法的触发执行应该在网络状态发生一定程度变化的时候, 所以端系统必须要能够利用某种机制及时了解网络运行情况, 比如经过一段时间的运行网络中有拥塞出现或者链路出现空闲等。显然如果没有网络层机制的支持算法的被触发和执行就将是盲目的, Max-Min 公平性的目标也很难达到, 通过一个视频流分层组播传输的实例来说明这一点, 如图 1。

当网络状态由 a 变为 b 时, 如果源 S 仍然采用图 1(a) 的

编码层次数和编码速率, 则对接收方 R₂ 来讲, 其接收速率虽然可以达到 1.0Mbps, 现在却只能获得 0.5Mbps, 显然降低了网络带宽利用率和接收方 R₂ 的视频接收质量, 造成网络带宽分配对 R₂ 不公平的现象。在某些链路上可用带宽下降的情况下也是如此。因此在实际的网络环境往往是动态的这一前提下, 传统的基于接收方的分层组播机制由于其源的层次划分是固定的, 显然将不利于提高网络带宽的利用率和整体视频接收质量, 造成网络中某些会话接收者被不公平对待的问题。基于此, 考虑能否根据网络中的拥塞状况对网络资源的空闲或紧缺的信息进行探测, 并在认为必要的时刻驱动基于效用的 Max-Min 公平速率分配算法的调用执行, 令发送方参与进行公平性资源分配, 确保通过系统的流量控制策略使整个网络资源被高效地加以利用。

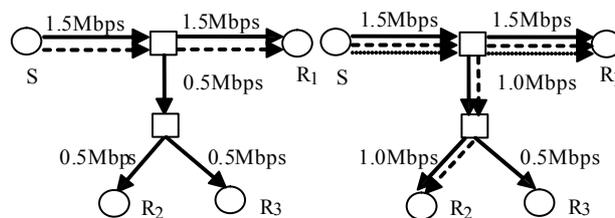


图 1 分层组播自适应

1.2 分层组播协议策略中的同步点问题

所谓同步点 SP^[3] (Synchronization Points) 就是指数据流

作者简介: 董春玲(1979 -), 女, 讲师、硕士, 主研方向: 组播拥塞控制, 多速率组播策略优化; 朱晓丽, 助教、硕士

收稿日期: 2006-02-13 **E-mail:** dongchunli@163.com

中带有特殊标记的数据包。同步点机制要求所有接收者只能在SP到来时才可以执行加入试验,且只能根据最近的SP以来的历史事件做出是否增加一层的决策。高层的SP个数比低层指数级递减,这样晚加入的接收者就可以很快赶上其它已经处在较高层的接收者,一段时间后共享同一瓶颈的接收者可以实现同步地加入或离开某些层。为降低加入事件失败的可能性,减少IGMP离开延时的不良影响,我们在同步点机制中加入Burst^[3]资源探测方法,即由发送者在SP到来之前发起一个临时的、短暂的数据突发周期(Burst)。在此期间每层上的速率翻倍,只有在Burst内不经历任何拥塞的接收者才被允许加入下一个更高的层次,这样就确保接收者只有在网络确实可能有充分带宽可用的情况下才能尝试加入一个高层。但Burst的发生也与同步点策略密切相关,所以合理设定各层上协调接收者加入和离开行为的同步点,包括同步点的疏密程度以及同步点间的时间间隔必将是关系分层组播流量控制性能优劣的重要因素。

图2是原始的RLC^[3]中同步点在各层上的设置情况。这种一成不变的设置方法并不能及时地反映网络状况,无论网络空闲与否,各层都会按照固定的周期执行加入试验、发起Burst探测,这在网络已经陷入拥塞的情况下可能会导致队列周期性溢出,加剧拥塞;而在网络空闲的情况下又可能因为既定的同步点间隔时间所限造成空闲带宽较长时间得不到利用的情况,这也正是RLC性能并不理想的原因之一。

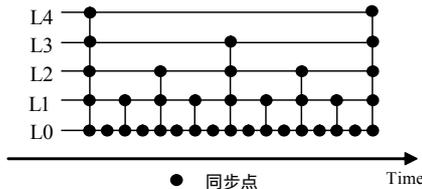


图2 RLC中的同步点

2 REM 算法辅助的分层组播流量控制方案的设计

为解决上述问题,引入主动队列管理算法REM^[4]作为在网络层获取显式拥塞指示、参与流量控制的途径,令接收端和源端流量控制相结合,分层组播协议根据运行REM的路由器上获得的拥塞指示信息进行流量控制的相应操作,推测和判断网络拥塞或空闲的程度,驱动分布式Max-Min公平速率分配算法的执行,由发送端计算各虚会话的公平资源分配速率,接收端据此主动调整其在组播树中的订阅层次;并利用路由器提供的拥塞信息动态调整各层上同步点的设置以及带宽探测Burst的突发周期,以期提高分层组播的流量控制和协议运行的性能,具体方案如下:

(1)在对网络状态的估测上,由运行REM的路由器按照一定的标记概率对到达的数据包进行标记或丢弃,当CE比特置位的数据包到达接收端后,接收方就在随后反馈给发送端的ACK报文中将ECN-Echo标志置位。这样发送端就可以统计一段时间内经过路由器被标记的数据包个数 $M_s(T)$ 与所发送数据包总个数 $F_s(T)$ 的比值即 $M_s(T)/F_s(T)$,称之为标记概率 $m_s(t)$,以此来估测链路上的丢失率情况。由于标记概率将作为评判网络状态的一个重要指标,因此 $m_s(t)$ 统计值的合理性和有效性对于流量控制相当关键。为了过滤由于偶然因素引起的 $m_s(t)$ 剧变,本文采用具有低通滤波效果的EMAW(Exponentially Weighted Moving Average)方法来保证丢失率(标记概率)的估计更加平滑,同时在一定程度上反映历

史丢包事件:

$$m_s(t) \leftarrow (1-\alpha)*m_s(t) + \alpha*m_s(T) \quad \alpha \in [0,1]$$

其中: $m_s(T)$ 为最近标记概率。为了及时反映网络变化状况, α 的取值可尽量大些(经仿真实验验证,当 α 取[0.6-0.8]时,对网络状态的估测效果较好)。

(2)在发送速率的调整策略上,发送端进一步统计一个时间段t内该发送端所属会话的标记概率均值 $m_s(t)_{ave}$ 的大小(将各个虚会话的统计值综合),统计方法为

$$m_s(t)_{ave} = \sum_{s \in S} m_s(t) / \text{Num}(S)$$

其中:Num(S)为会话S中包含虚会话的总个数。然后将其与历史时间段内统计的值进行比较,如果 $|m_s(t)_{ave} - m_s(t-1)_{ave}| / m_s(t-1)_{ave} > \Delta$,则认为第t时间段较之于第t-1时间段内网络状态发生较大变化,网络可能出现明显拥塞或者出现明显空闲,认为有必要重新为各虚会话分配带宽,那么驱动分布式Max-Min公平速率分配计算算法Algorithm1的运行,结合各接收者的链路状态以及速率请求,计算得出各虚会话的公平速率集RateSet D。然后将取值于连续数据集的RateSet D交付Algorithm 2,以BINL(RateSet D, real δ)进行离散化分层,最后由接收者根据自己在新的速率集RateSet中所处的层次,选择加入新层或者离开当前所在层,或者保持不变。这里 t 和 Δ 为可配置参数,目的在于考虑不同应用对速率振荡的要求不同,比如实时媒体分发就要求尽量避免剧烈速率振荡,此时可将 Δ 设置得略微小一些;而对于数据分发,则可将 Δ 酌情设置得小一些,但也不能太小,以免造成网络频繁抖动(这里Algorithm 1和Algorithm 2见文献[2])。

(3)在同步点的设置上,由发送方依据路由器提供的拥塞指示 $m_s(t)$ 的大小,对各层同步点之间的时间间隔长度(时隙time slot)进行调整,这种调整将是随网络状态而动态变化的,而不是RLC的一成不变,以避免在网络并没有可用带宽的情况下依然盲目地进行Burst探测或者Join Experiment,导致定期的队列溢出和不必要的网络拥塞加剧。我们的办法是在多速率组播会话启动时为各层上同步点出现的时间间隔设定一个初始值 $TS(t_0)$,该值的设定依照各层上同步点个数指数级减少的宗旨以便于令处于低层的接收者获得较多的加入试验机会,然后随着网络运行,在网络负载状况发生较大变化时,即 $|m_s(t)_{ave} - m_s(t-1)_{ave}| / m_s(t-1)_{ave} > \Delta$ 时,各层同步点间的时间间隔 $TS(t)$ 均在 $TS(t_0)$ 基础上变化 $V(m_s(t))$ 倍,即: $TS(t) = TS(t_0) * V(m_s(t))$ 。我们选取调节函数 $V(x)$ 为 $x \in [0,1]$ 、 $V(x) \in [\beta, +\infty)$ 的单调递增函数,形式如图3所示。

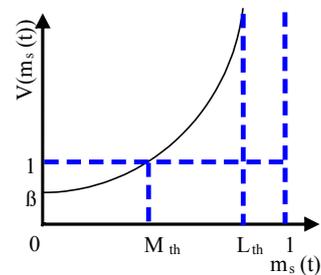


图3 同步点间隔调节函数

其中, M_{th} 、 L_{th} 为给标记概率值 $m_s(t)$ 设定的两个阈值($0 < M_{th} < L_{th} < 1$), M_{th} 的值不应过大,一般都小于0.02。当 $m_s(t)$ 的取值在 $[0, M_{th})$ 间时,认为网络基本不拥塞,有空闲带宽可用,那么就可以适当减小同步点间的时间间隔 $TS(t)$,

即令 $V(m_s(t))$ 取小于1的值,但也不应无限地缩小,所以设定了下届控制参数 ($0 < \beta < 1$, 为可调参数);当 $m_s(t) = M_{th}$ 时, $V(m_s(t))=1$,即 $TS(t)$ 的值保持 $TS(t_0)$ 不变;当 $m_s(t) > M_{th}$ 时, $TS(t)$ 被放大,即令 $V(m_s(t))$ 取大于1的值;随 $m_s(t)$ 取值逐渐增大,甚至超过 L_{th} 时,认为网络拥塞程度比较严重了,那么就令 $V(m_s(t))$ 取无限大的值,基本取消加入试验的同步点,以免造成网络拥塞的加剧。根据上述分析,取同步点间隔调节函数为

$$V(m_s(t)) = \begin{cases} \frac{1-\beta}{M_{th}^2} (m_s(t))^2 + \beta, & \text{当 } m_s(t) \in [0, L_{th}] \text{ 时} \\ +, & \text{当 } m_s(t) \in (L_{th}, 1] \text{ 时} \end{cases}$$

由于 Burst 探测都是在同步点到来之前进行的,因此 Burst 的周期间隔也就随着同步点的设置策略得到了优化。这样就实现了我们所预期的目标:当网络中并无空闲带宽的情况下,应减少接收者盲目进行加入试验或 Burst 探测带宽事件的可能性(增大 $TS(t)$ 的长度),避免不必要的数据包丢失和周期性网络拥塞;而在链路带宽出现空闲时,又适当增加加入试验或 Burst 的频率(减小 $TS(t)$ 的长度),将该同步点机制作为流量控制策略的重要组成部分,以确保分层组播协议的灵敏性和网络资源利用的充分性。

3 仿真实验

我们以 Red Linux 8.0 作为平台,使用伯克利大学的网络研究组 LBNL 研制开发的网络仿真器 NS 作为仿真实验环境,分别就稳定性、灵敏性等 4 方面进行实验。本文所设计的方案称为 FLM。

(1)稳定性:采用图 4 所示网络拓扑,在 FLM 与 RLC、RLM^[5] 3 种协议的稳定性之间进行比较,图 5 是三者完全等同的网络条件下的实验结果。可以看到 FLM 在逐层加至最大可能层之后,只进行了一次 Burst 就维持了平稳运行速率,整个网络状态基本稳定,与 RLC 和 RLM 形成对照。

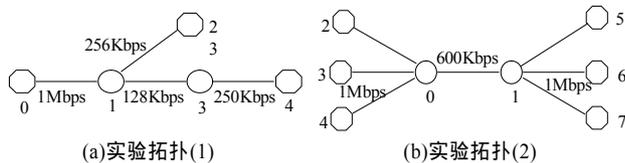


图 4 网络拓扑

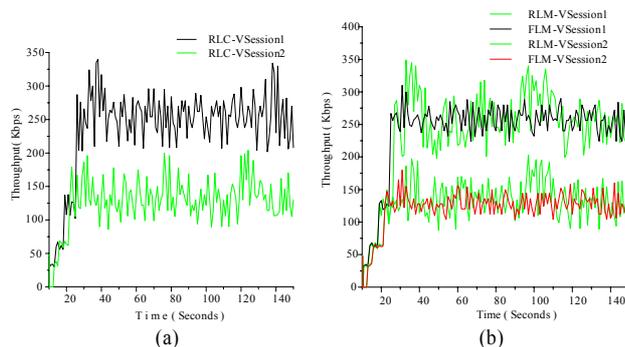


图 5 稳定性试验

(2)灵敏性:在上述实验条件下,令拓扑图 4 中链路 1-2 上的带宽在 100s~150s 期间变为 128kbps, 150s~200s 又变回 256kbps,以 2 号接收者为观察目标。结果如图 6 所示,FLM 组播会话能对链路带宽的变化做出积极而迅速的响应。

(3)丢失率:在灵敏性实验的过程中统计 flow1 在节点 1 处的丢失率的变化情况,并与 RLM、RLC 在同等条件下进行比较,如图 7 示。FLM 所出现的丢包多是 IGMP 离开延时过

程中的短暂丢失,总体丢失率比 RLM 少得多。至于 RLC,它是在 RLM 基础上做的改进,但是 A.Legout 等人对其进行的测试也证明并没有预料中的优良性能,反而表现出许多缺陷比如没有解决收敛速度相关的问题,丢失率也比较高,通常在 2%~15%之间。

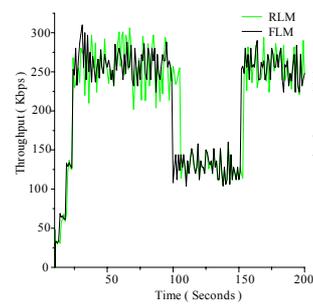


图 6 灵敏性试验

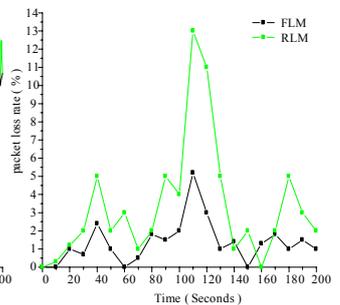


图 7 丢失率试验

(4)公平性:采用图 5 所示拓扑,在组播会话与 TCP 会话共存的情况下进行实验,结果如图 8 所示。通过比较容易看出 FLM 显示出较好的 TCP 友好特性,在瓶颈链路上出现拥塞后 FLM 能主动降低订阅层数,保证对 TCP 的公平性,而且这种降低是 TCP-Like 的指数级下降,对于快速解除网络拥塞也具有很好的效果。

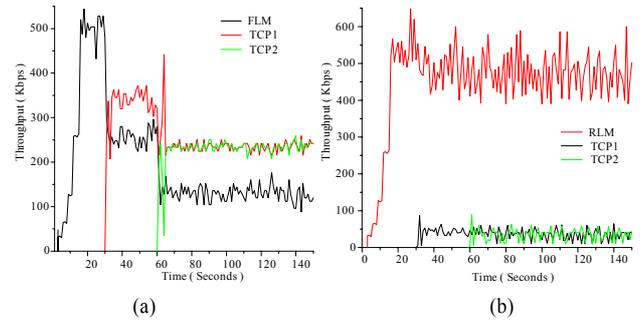


图 8 公平性试验

4 总结

本文设计了一种基于速率的、由接收者和发送者混合驱动的分层多速率组播流量控制方案,该方案结合路由器上的主动队列管理算法,通过 REM 算法所提供的网络状态指示信息来触发 Max-Min 公平速率分配算法的运行,并对现有多速率组播解决办法在协调接收者行为的同步点策略上进行改进。本文通过一系列仿真实验来评价算法的性能,与分层组播协议中有代表性的 RLM、RLC 作了详细的比较,结果表明本方案具有良好的公平性特征,提高了灵敏性,在稳定性和收敛性方面均有较大改善。

参考文献

- 1 Rubenstein D, Kurose J, Towsley D. The Impact of Multicast Layering on Network Fairness[C]//Proc. of ACM SIGCOMM'99, 1999.
- 2 董春玲, 朱晓丽, 郑明春. 分层组播中 MAX-MIN 公平速率分配算法的运用[J]. 计算机应用, 2005, 25(3): 533-535.
- 3 Vicisano L, Rizzo L, Crowcroft J. TCP-like Congestion Control for Layered Multicast Data Transfer[C]//Proc. of IEEE INFOCOM, 1998: 996-1003.
- 4 Athuraliya S, Low S, Li V, et al. REM: Active Queue Management[J]. IEEE Network Magazine, 2001, 15(3): 48-53.
- 5 McCanne S, Jacobson V, Vetterli M. Receiver-driven Layered Multicast[C]//Proc. of ACM SIGCOMM, Palo Alto, CA, USA, 1996: 117-130.