

区分服务网络中主动队列管理算法*

刘威¹⁺, 程文青¹, 何建华¹, 乐春晖¹, CHOU Chun-Tung², 杨宗凯¹

¹(华中科技大学 电子与信息工程系,湖北 武汉 430074)

²(School of Computer Science and Engineering, The University of New South Wales, Sydney NSW 2052, Australia)

Active Queue Management Algorithms in DiffServ Networks

LIU Wei¹⁺, CHENG Wen-Qing¹, HE Jian-Hua¹, LE Chun-Hui¹, CHOU Chun-Tung², YANG Zong-Kai¹

¹(Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

²(School of Computer Science and Engineering, The University of New South Wales, Sydney NSW 2052, Australia)

+ Corresponding author: Phn: +86-27-87450745, Fax: +86-27-87540745, E-mail: liuwei@mail.hust.edu.cn, http://itec.hust.edu.cn

Received 2003-09-16; Accepted 2004-03-17

Liu W, Cheng WQ, He JH, Le CH, Chou CT, Yang ZK. Active queue management algorithms in DiffServ networks. *Journal of Software*, 2005,16(6):1120-1130. DOI: 10.1360/jos161120

Abstract: Current assured forwarding (AF) service in differentiated services (DiffServ) networks can provide stable guarantees in throughput, but is lacking of efficient schemes in ensuring queuing delay and loss ratio. By analyzing the steady state operating point of RIO, this paper proposes two active queue management algorithms with adaptive control policy, namely ARIQ-D and ARIQ-L. These two algorithms can provide differentiated performance in, respectively, queuing delay and loss ratio, in addition to throughput guarantee. By deploying ARIQ-D and ARIQ-L, AF service can provide quantitative guarantees for multimedia traffic with multiple QoS metrics.

Key words: differentiated service; active queue management; quantitative assured forwarding; QoS; RIO

摘要: 现有区分服务网络的保证转发服务可提供稳定的带宽保证,但缺乏保证时延和分组丢失性能的有效方案.基于对RIO队列的稳态性能分析,提出两种自适应调整控制策略的主动队列管理算法(ARIO-D和ARIO-L).仿真结果表明,这两种算法在保持RIO算法带宽保证能力的同时,还可以提供稳定的和可区分的时延和分组丢失性能.采用ARIO-D和ARIO-L的保证转发服务可以为多媒体流量提供多种服务质量的定量保证.

关键词: 区分服务;主动队列管理;定量保证转发;服务质量;RIO

* Supported by the National Natural Science Foundation of China under Grant No.60202005 (国家自然科学基金); the Foundation for the Key Teacher of the Ministry of Education of China (国家教育部骨干教师基金); the Foundation of Australian Research Council under Grant No.LX0240468 (澳大利亚研究院基金)

作者简介: 刘威(1977-),男,湖北武汉人,博士,讲师,主要研究领域为网络服务质量,多媒体内容分发技术;程文青(1964-),女,副教授,主要研究领域为高速通信网络,多媒体网络;何建华(1976-),男,博士,副教授,主要研究领域为无线多媒体传输技术;乐春晖(1976-),男,博士生,主要研究领域为网络服务质量,接入控制;CHOU Chun-Tung(1967-),男,博士,Senior Lecturer,主要研究领域为流量工程,内容分发网络;杨宗凯(1963-),男,博士,教授,博士生导师,主要研究领域为高速通信网络,网络教育标准和技术.

中图法分类号: TP393 文献标识码: A

区分服务(differentiated service,简称 DiffServ)^[1]被视为解决未来 IP 网络服务质量(quality of service,简称 QoS)问题的主要解决方案.区分服务中,端到端(end-to-end)的服务质量保证任务被分解到边缘到边缘(edge-to-edge)的网络域和单个网络节点上,分别由 PDB(per domain behavior)^[2]和 PHB(per hop behavior)^[1]描述和规范.在已定义的 PHB 中,保证转发(assured forwarding,简称 AF)^[3]可以提供可区分的多个服务级别服务质量保证,与加速转发(expedited forwarding,简称 EF)^[4]相比,资源利用率更高.

目前,实现保证转发的带宽保证的研究已经比较成熟,常见的方案^[5]是在边缘节点部署 TSW(time sliding window)标记算法,依据订购速率对到来流量标记为订购内(IN)和订购外(OUT),在内部节点采用 RIO(RED with IN/OUT)算法,对 IN 和 OUT 队列进行有区分的处理.但是,在时延和分组丢失率性能保证方面,却缺乏良好的解决方案.由于时延和分组丢失的性能需求,信息不便于通过包头标记所携带,早期的研究偏重于在单节点上提供定性的、相对的区分服务,如 Dovrolis 提出的按比例区分服务方案^[6],以特定的调度算法和队列管理算法实现单节点上的按比例时延和分组丢失性能,再以动态的服务级别管理机制协调沿途各节点,最后达到边缘到边缘定量的性能保证.这种按比例区分服务方案的缺点是需要多节点甚至多域之间进行较多信令协调和接入控制.最近的研究倾向于在单节点上提供定量的区分服务,如 Christin 提出的定量保证转发方案^[7],先以启发式算法获得符合需求的各服务级别的服务速率,再以反馈控制的方法调整队列调度和丢弃动作,分别实现时延和分组丢失率的控制.该定量保证转发方案的缺点是实现复杂度高,需要启发式算法来估计速率,并两次尾部丢弃来满足比例区分性和定量性.这些方案^[6,7]虽然可以在保证转发服务的框架上实现,但是部署的复杂度都比较大,总体扩展性有限.

为实现保证转发上多种性能(带宽,时延和分组丢失)的服务质量保证,在单节点上提供定量的区分服务是一种简单、有效的方案.PHB 性能主要通过队列管理算法或调度算法实现.如果可以在现有 RIO 算法的基础上,通过改进队列管理算法实现单节点路由器上的可区分的时延和分组丢失性能,则所构成的新的保证转发服务具有重大意义.一方面,新的保证转发服务可以由现有 TSW+RIO 方案扩展而成,便于部署和实现;另一方面,新的保证转发服务可以对多种服务质量指标提供灵活 PHB 性能,更加适合多媒体流量的传输.为此,本文选择 RIO 算法作为改进对象,提出两种自适应 RIO 算法(ARIO-D 和 ARIO-L),分别适用于实现单节点上的时延和分组丢失的定量的服务质量保证.

1 研究背景和研究目标

1.1 区分服务中的主动队列管理

主动队列管理(active queue management,简称 AQM)是目前队列管理的主流技术,该研究源于拥塞控制领域,其中 RED 算法^[8]是最主要的 AQM 算法.区分服务网络中的 AQM 除了支持拥塞控制,还需要实现服务质量保证.其研究思路主要是在原 AQM 算法基础上扩展实现对 IN 和 OUT 流量的不同处理,如从 RED 算法扩展得到的 RIO 算法,以及对 PI 算法和 REM 算法的扩展^[9,10].RIO(RED with IN/OUT)算法包括两个 RED 算法实例分别处理订购内和订购外流量,在通常情况下可以获得较高的吞吐量和较低的时延.在保证转发服务可采用的 AQM 中,RIO 算法比其他算法更易于实现,如 PI 算法在保证转发中的扩展实现需要实时调整边缘的标记速率以实现带宽保证^[9],而 REM 算法在保证转发中的扩展实现需要端到端的用户反馈控制支持^[10].因此,本文采用 RIO 算法作为研究和改进的对象.

RIO 算法的工作过程可以简单描述为:以 $(th_{\min}^{\text{in}}, th_{\max}^{\text{in}}, p_{\max}^{\text{in}})$ 格式的参数字对表示 RED 算法中的最小分组丢失队列门限,最大分组丢失队列门限和最大分组丢失概率的参数.RIO 算法需要分别配置 IN 流量的 RED 参数 $(th_{\min}^{\text{in}}, th_{\max}^{\text{in}}, p_{\max}^{\text{in}})$ 和 OUT 流量的 RED 参数 $(th_{\min}^{\text{out}}, th_{\max}^{\text{out}}, p_{\max}^{\text{out}})$.本文中的 RIO 的参数配置满足两个 RED 的队列处理部分不重叠的条件,即 $th_{\min}^{\text{out}} < th_{\max}^{\text{out}} \leq th_{\min}^{\text{in}} < th_{\max}^{\text{in}}$.在 RIO 运行过程中,如果整体平均队列长度 avg 超过 th_{\min}^{out} ,OUT 分组将以概率 $(avg - th_{\min}^{\text{out}}) / (th_{\max}^{\text{out}} - th_{\min}^{\text{out}}) \cdot p_{\max}^{\text{out}}$ 被随机丢弃;如果 avg 继续增长并超过 th_{\max}^{out} ,则 OUT 分

组将以概率 1 被丢弃,同时 IN 分组也将开始以概率 $(avg^{in}-th_{min}^{in})/(th_{max}^{in}-th_{min}^{in}) \cdot p_{max}^{in}$ 被随机丢弃,其中 avg^{in} 是 IN 分组队列的平均队列长度.

1.2 研究目标和研究场景

服务质量的保证需求可以分解为稳定性和服务可区分性两个方面^[11].本文对 RIO 算法的改进目标也是如此,即实现时延和分组丢失率性能的稳定性和服务可区分性.由于区分服务网络节点不保留每个流的信息状态,同一源节点发出的分组有可能被先后标记为 IN 或者 OUT,仅仅保证 IN 分组不丢失并不意味着该源节点没有分组丢失.对那些 QoS 要求较高的多媒体流量而言,IN 和 OUT 流量的时延和分组丢失都应该尽量减小.因此,本文采用的评价指标包括订购流量的吞吐率,队列整体(包括 IN 和 OUT)的排队时延和分组丢失率等.

考虑到实际网络的设计运行负载都不高,本文着重讨论未完全订购(under-subscription)情况下的区分服务网络,即 RIO 算法主要工作于处理 OUT 分组的 RED 算法部分.本文假设通过区分服务域中的主要流量为 TCP 流量.随着 TCP 友好(TCP-friendly)的速率控制技术(如 TFRC^[12])在非弹性流量(如 UDP)发送源的部署,本文在 TCP 流量下的分析结果也可以推广到其他流量.在前期工作^[13]中,我们验证了 TFRC 流量在区分服务中与 RIO 算法交互时呈现与 TCP 流量类似的性能.

2 稳态 RIO 队列的建模分析

2.1 TCP聚合流在RIO队列中的稳态性能

区分服务框架下相同服务等级的 TCP 流被聚合为 TCP 聚合流接受统一处理.由文献[14]中提供的区分服务域内的 TCP 吞吐量公式,可以得到由 n 个 TCP 流构成的聚合流的吞吐量为

$$B_A = \frac{3}{4}S + \frac{3k}{4} \cdot \sum_{i=1}^n \frac{1}{RTT_i} \sqrt{\frac{2}{p_i^{out}}} \quad (1)$$

其中 B_A 表示 TCP 聚合流的吞吐量, S 表示该链路的订购带宽, k 表示平均 TCP 包长度, RTT_i 表示第 i 个 TCP 流的平均往返时间(round trip time,简称 RTT); p_i^{out} 表示第 i 个 TCP 流中 OUT 包的平均分组丢失率.

为简化起见,我们假设聚合流中的各 TCP 流都有相同的平均往返时间和分组丢失率,设 $RTT_i=RTT$ 和 $p_i^{out}=p^{out}$,则式(1)可以改写为

$$B_A = \frac{3}{4}S + \frac{3kn}{4RTT} \cdot \sqrt{\frac{2}{p^{out}}} \quad (2)$$

假设 RIO 算法工作在稳态时,该内部节点的链路利用率为 100% (即 $B_A=C$),则边缘节点的标记概率 p_{mark} 可以表示为

$$p_{mark} = \frac{S}{B_A} = \frac{S}{C} \quad (3)$$

另一方面,TCP 往返时间 RTT 包括了链路传输时延和队列处理时延,设平均传输时延为 R ,设 RIO 算法的稳态队列长度为 q ,则有:

$$RTT = R + \frac{q}{B_A} \quad (4)$$

由式(2)、式(3),我们可以得到 RIO 队列的稳态队列长度的表达:

$$q = \frac{kn}{4/3-S/C} \cdot \sqrt{\frac{2}{p^{out}}} - CR \quad (5)$$

进一步还可以获得由此决定的 RIO 稳态系统的整体队列时延 $delay$ 和分组丢失率 $loss$,性能分别为

$$delay = \frac{k^2n}{4/3 \cdot C - S} \cdot \sqrt{\frac{2}{p^{out}}} - kR \quad (6)$$

$$loss = p^{in} \cdot p_{mark} + p^{out} \cdot (1 - p_{mark}) \quad (7)$$

该简单模型给出了 RIO 系统在稳定状态下的主要性能指标的平均值.文献[15]对 TCP-RED 系统进行了类似的建模工作,但是与本文的研究场景不同.本文的建模需要考虑区分服务网络中 TCP 在聚合后的整体行为特征,以及不同的吞吐量订购比例(S/C),采用式(2)描述通过 RIO 系统的聚合流吞吐量;而文献[15]仅考虑单个 TCP 流通过 RED 系统的情形,采用文献[16]中的公式描述 TCP 吞吐量.

另外,需要指出的是,虽然该模型中的网络流量负载 n 是以理想化的同构 TCP 流数目来衡量的,与实际网络中的异构 TCP 流或者 TCP/UDP 混合流量情况有一定的区别,但这并不妨碍我们的分析.我们关注的是 RIO 队列在统计意义上的平均性能,而非异构流量的公平性问题,因此可以忽略异构流量之间的差异,仅用 n 来表征等效 TCP 负载流数目,视其数值反映了异构 TCP 流窗口长度、异构 TCP 的 RTT 以及与 UDP 流量等多种因素的作用结果.

2.2 RIO队列的稳态性能分析

文献[15]提出了一种分析 RED 算法稳态性能和稳定性的方法,如图 1(a)所示.用 $G(p)$ 表示 RED 队列中平均队列长度关于分组丢失率的函数,对应图中队列规则曲线(queue law);用 $H(q)$ 表示 RED 算法中根据平均队列长度计算分组丢失概率的函数,对应图中控制规则曲线(control law).文献[15]证明了 RED 队列的稳定条件是这两条曲线在 RED 的随机丢包阶段有交点,即存在 (p_s, q_s) 且满足 $(0 < p_s < p_{max}, th_{min} < q_s < th_{max})$.该交点 (p_s, q_s) 是一个吸引子,即在网络条件没有变化的情况下,一旦 RED 队列系统达到该点,则将稳定于该点附近并作小范围的波动.

在本文讨论的区分服务网络域中,因为订购比例未达到饱和,则理想情况下没有 IN 包被丢弃,RIO 算法主要工作于处理 OUT 包部分的 RED.对由整体平均队列长度和 OUT 分组丢失率构成的平面曲线系统而言,上述稳定性条件依然成立.文献[17]的仿真实验证明了该模型在区分服务域中的有效性.本文将用此方法讨论 RIO 队列系统的稳定性问题.取式(5)作为 RIO 队列系统的 $G(p)$,取 OUT 部分 RED 计算分组丢失概率的公式:

$$p^{out} = \frac{q - th_{min}^{out}}{th_{max}^{out} - th_{min}^{out}} \cdot p_{max}^{out} \tag{8}$$

作为 RIO 队列系统的 $H(q)$,可以作出 RIO 系统如图 1(b)所示.

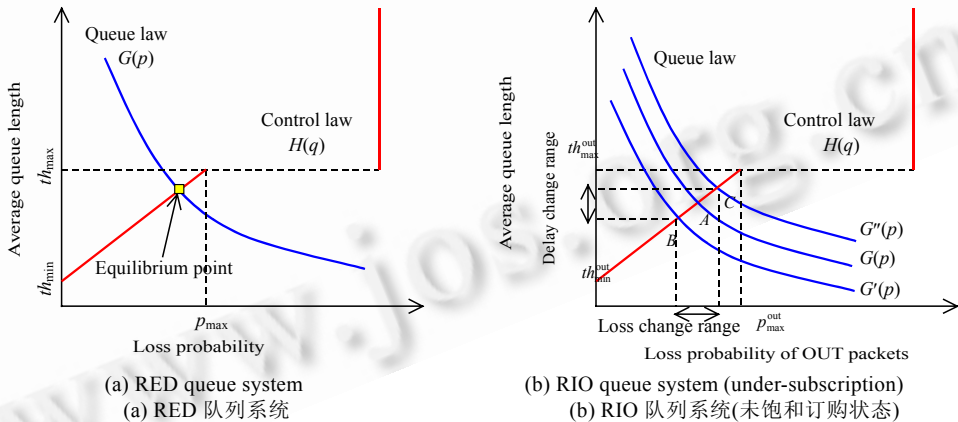


Fig1 Queue law and control law

图 1 队列规则曲线和控制规则曲线

(1) RIO 算法的稳定性分析

由式(5)可知,影响 RIO 队列稳态队列长度 q 的因素包括:网络流量负载(以 TCP 流数目衡量的 n),链路带宽订购比例(S/C ,或者边缘路由器的标记概率 p_{mark}),TCP 流往返时间(RTT),OUT 包的平均分组丢失率(p^{out}).在这些因素中, n 是与实际网络条件有关的可能发生动态变化的量, S/C 是实际网络中可能被管理者重新配置的量,当它们发生变化时,RIO 的队列规则曲线 $G(p)$ 将发生整体移动.如果队列规则曲线与控制规则曲线无交点,则系统工作在不稳定的状态,各性能都将受到损失;如果两曲线获得新的交点,如图 1(b)中的 $G'(p)$ 或者 $G''(p)$ 所示,系统将工作于新的稳定点 B 或者 C,此时的 IN 流量吞吐量保证虽然没有影响,但是整体队列的时延和分组丢失性能将

出现图中所示的波动.因此,RIO 系统的性能会随着网络负载变化而有所不同,这种性能的不稳定性不利于服务质量的保证.提高 RIO 算法稳定性的直观想法是,通过动态调节 OUT 包的平均分组丢失率 p^{out} 调整控制规则曲线,从而使系统在期望区间内重新获得稳定的工作点.

(2) RIO 算法的服务可区分能力分析

当 RIO 算法的队列规则曲线 $G(p)$ 发生移动之后,一旦 RIO 系统获得新的稳定状态,此时订购内流量仍可获得 100% 的订购带宽,但是队列的整体时延和分组丢失率性能则会因稳定点的不同而发生变化.因此,RIO 算法仍具备带宽性能服务可区分的能力,却不具备时延或者分组丢失性能服务可区分的能力.如果我们提出的改进算法可以控制图 1(b)中 $G(p)$ 和 $H(q)$ 的交点区间,使得新算法可以保证时延或者分组丢失性能,则新算法将具备服务可区分的能力.

3 自适应的 RIO 算法

3.1 自适应RIO算法的设计

我们借鉴自适应 RED(adaptive RED)算法^[18,19]的思想,提出如图 2 所示的两种自适应的方法来调整 RIO 算法的控制规则曲线,以达到两曲线在期望区间相交的目标.

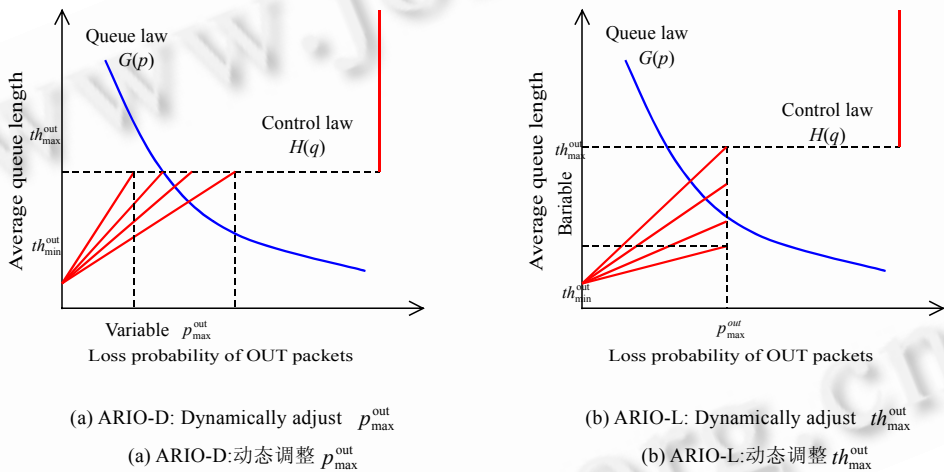


Fig.2 Dynamically adjust the control law of OUT part RIO

图 2 动态调整 RIO 算法的 OUT 部分的控制规则曲线

3.1.1 ARIO-D 算法

如图 2(a)所示,保持 OUT 包的最大分组丢失队列门限 th_{max}^{out} 不变,动态调整 OUT 包的最大分组丢失概率 p_{max}^{out} 来满足两曲线的相交条件.该自适应方法以某个队列区间为控制目标,可以保证不同或者动态的网络条件下的队列时延性能,适用于时延敏感(delay-sensitive)或者时延抖动敏感(jitter-sensitive)的网络服务(如点播流媒体业务).本文称之为考虑时延的自适应 RIO 算法,简称 ARIO-D(adaptive RIO for delay).

3.1.2 ARIO-L 算法

如图 2(b)所示,保持 OUT 包的最大分组丢失概率 p_{max}^{out} 不变,动态调整 OUT 包的最大分组丢失队列门限 th_{max}^{out} 来满足两曲线的相交条件.该自适应方法以某个区间的平均分组丢失率为控制目标,可以保证 RIO 算法在不同或者动态网络条件下的分组丢失性能,适用于对分组丢失敏感(loss-sensitive)的网络服务(如保密数据传输业务).本文称之为考虑分组丢失的自适应 RIO 算法,简称 ARIO-L(adaptive RIO for loss).

3.2 自适应RIO算法的实现

RIO 算法的自适应控制策略在原 RIO 算法的基础上改造实现.在每个自适应调整的周期内,通过指数加权滑动平均(exponentially weighted moving average,简称 EWMA)的方法估算系统的平均队列长度或者平均分组丢失率,然后将这些参量的变化与事先定义的系统期望值比较以获得控制方向,最后通过调整 OUT 包的最大分

组丢失概率(p_{\max}^{out})或者最大分组丢失队列门限(th_{\max}^{out})来调整系统的队列性能,以达到稳定性控制的目标。

算法 1 和算法 2 分别给出了 ARIO-D 和 ARIO-L 的实现算法。考虑到最大分组丢失概率 p_{\max}^{out} 的取值范围是 (0,1), 乘性减少有利于调整 p_{\max}^{out} 趋近于 0, 因此, ARIO-D 算法的自适应调整部分采用了加性增加乘性减少 (additive increase multiplicative decrease, 简称 AIMD) 的方法, p_{\max}^{out} 的调整区间为 [0.01, 0.5]。考虑到最大分组丢失队列门限 th_{\max}^{out} 在缓存区内的调整范围分布广泛, 加性增加和减少有利于均匀覆盖其变化区间, 所以, ARIO-L 算法的自适应调整部分采用了加性增加、加性减少 (additive increase additive decrease, 简称 AIAD) 的方法, th_{\max}^{out} 的调整区间为 $[th_{\min}^{\text{out}}, th_{\min}^{\text{in}}]$ 。

算法 1. ARIO-D 算法.

Every adapting interval:

if ($q > \text{target}$ and $p_{\max}^{\text{out}} < 0.5$)

Increase $p_{\max}^{\text{out}} : p_{\max}^{\text{out}} + \alpha$

else if ($q < \text{target}$ and $p_{\max}^{\text{out}} > 0.01$)

Decrease $p_{\max}^{\text{out}} : p_{\max}^{\text{out}} \times \beta$

Variables:

q : average queue length estimated by EWMA

q' : current measured queue length

weight : weight value in EWMA

$q = q' \times \text{weight} + q \times (1 - \text{weight})$

Parameters:

target range: $[\gamma \times (th_{\max}^{\text{out}} + th_{\min}^{\text{out}}) \pm \eta \times (th_{\max}^{\text{out}} - th_{\min}^{\text{out}})]$

α and β : increment and decrease factor

γ and η : position factors to determine target queue range

算法 2. ARIO-L 算法.

Every adapting interval:

if ($p > \text{target}$ and $th_{\max}^{\text{out}} < th_{\min}^{\text{in}}$)

Increase $th_{\max}^{\text{out}} : th_{\max}^{\text{out}} + \alpha$

else if ($p < \text{target}$ and $th_{\max}^{\text{out}} > th_{\min}^{\text{out}}$)

Decrease $th_{\max}^{\text{out}} : th_{\max}^{\text{out}} - \beta$

Variables:

p : average loss ratio of OUT packets estimated by EWMA

p' : current measured loss ratio of OUT packets

weight : weight value in EWMA

$p = p' \times \text{weight} + p \times (1 - \text{weight})$

Parameters:

target range: $[\gamma \times p_{\max}^{\text{out}} \pm \eta]$

α and β : increment and decrease factor

γ and η : position factors to determine target loss range

3.3 自适应RIO算法的参数配置

3.3.1 ARIO-D 算法参数的设置

(1) 目标控制参数 γ 和 η

γ 和 η 是实现算法对时延性能定量控制的主要参数,其配置需要区分服务域管理员结合本节点需要提供的时延保证指标计算得到.设 d 和 j 表示在该节点上期望得到的时延以及时延抖动的值,该网络节点分配给RIO队列的带宽为 C ,分组的平均大小为 k .因为ARIO-D算法提供的期望平均队列长度值为 $\gamma(th_{\max}^{\text{out}} + th_{\min}^{\text{out}})$,以队列长度衡量的最大时延抖动值为 $2\eta(th_{\max}^{\text{out}} - th_{\min}^{\text{out}})$ (以队列长度来衡量),所以 γ 和 η 可以由下面两式确定:

$$\gamma = \frac{C \cdot d}{k \cdot (th_{\max}^{\text{out}} + th_{\min}^{\text{out}})}, \eta = \frac{C \cdot j}{2k \cdot (th_{\max}^{\text{out}} - th_{\min}^{\text{out}})} \quad (9)$$

(2) 自适应调节参数 α 和 β

α 和 β 控制每次自适应调整 p_{\max}^{out} 的幅度,其配置应当满足以下稳定性条件:单次对 p_{\max}^{out} 调节所导致的队列波动 Δq 应当小于最大队列抖动值 $2\eta(th_{\max}^{\text{out}} - th_{\min}^{\text{out}})$.而 α 和 β 所导致的最大队列波动 $\Delta q(\alpha)$ 和 $\Delta q(\beta)$ 分别由下列两式给出:

$$\Delta q(\alpha) = \left(\frac{p_{\max}^{\text{out}}}{p_{\max}^{\text{out}} - p_{\max}^{\text{out}} + \alpha} - \frac{p_{\max}^{\text{out}}}{p_{\max}^{\text{out}} + \alpha} \right) \cdot (th_{\max}^{\text{out}} - th_{\min}^{\text{out}}) \leq \frac{\alpha}{p_{\max}^{\text{out}} + \alpha} \cdot (th_{\max}^{\text{out}} - th_{\min}^{\text{out}}),$$

$$\Delta q(\beta) = \left(\frac{p_{\max}^{\text{out}}}{p_{\max}^{\text{out}} \cdot \beta} - \frac{p_{\max}^{\text{out}}}{p_{\max}^{\text{out}}} \right) \cdot (th_{\max}^{\text{out}} - th_{\min}^{\text{out}}) \leq \frac{1 - \beta}{\beta} \cdot (th_{\max}^{\text{out}} - th_{\min}^{\text{out}}).$$

因此,可以得到满足ARIO-D算法稳定条件的 α 和 β 范围如下:

$$0 < \alpha \leq \frac{2\eta}{1 + 2\eta}, \quad \frac{1}{1 + 2\eta} \leq \beta < 1 \quad (10)$$

3.3.2 ARIO-L 算法参数的设置

(1) 目标控制参数 γ 和 η

与ARIO-D算法一样, γ 和 η 的配置需要网络管理员结合本节点需要提供的分组丢失保证指标计算得到. η 的设置与期望的分组丢失性能控制精度有关,常见的数值为0.005左右.设 l 表示在该节点上期望得到的分组丢失率的值,因为ARIO-L算法提供的期望分组丢失率为 $p^{\text{out}} = \gamma p_{\max}^{\text{out}}$,结合式(7)和 $p^{\text{in}} = 0$ 的理想情况假设,则 γ 可以由下式确定.

$$\gamma = \frac{l}{(1 - p_{\text{mark}}) p_{\max}^{\text{out}}} \quad (11)$$

(2) 自适应调节参数 α 和 β

与ARIO-D算法一样, α 和 β 的配置应当满足自适应调整的稳定性条件:单次对 th_{\max}^{out} 所导致的分组丢失率的波动 Δp 应当小于设置的最大分组丢失率的抖动值 2η .类似于在ARIO-D算法中的推导,可以得到满足ARIO-L算法稳定条件的 α 和 β 范围如下:

$$0 < \alpha, \beta \leq \frac{2\eta}{p_{\max}^{\text{out}} + 2\eta} (th_{\max}^{\text{out}} - th_{\min}^{\text{out}}) \quad (12)$$

3.4 自适应RIO算法的部署

自适应RIO算法可以为保证转发服务提供可区分的时延和分组丢失性能.在传输多媒体流量时,一种可能的部署方案是将不同的多媒体应用映射到不同的服务级别中去.例如,直播的流媒体应用比点播的流媒体应用需要更为严格的时延,我们可以将它们映射到不同的ARIO-D的队列中去.又例如,保密数据通信对于分组丢失率的要求比普通数据服务要高,可以设置到不同的ARIO-L的队列中去.另一种可能的方案是结合分层编码技术,按照同一个流媒体中不同子层的传输解码需求,将其映射到不同的服务级别中去,例如,我们可以通过文献[20]中给出的时延分配准则在多个保证转发服务中配置ARIO-D算法.

4 仿真与结果

为考察自适应RIO算法在服务质量保证方面的能力,我们设计了两组实验比较其与原RIO算法在稳定性和服务可区分性方面的能力.仿真实验环境为ns-2^[21],网络拓扑及主要配置如图3所示.在哑铃型的区分服务网络域中,edge1和edge2分别为边缘节点,core为内部节点.从src_i到dst_i($i=1,2,\dots,N$)建立 N 个TCP的连接.core

到 edge2 的链路为瓶颈链路,入口边缘节点 edge1 安装 TSW 的标记器,内部节点 core 上分别安装 RIO,ARIO-D 和 ARIO-L 这 3 种算法并运行仿真。

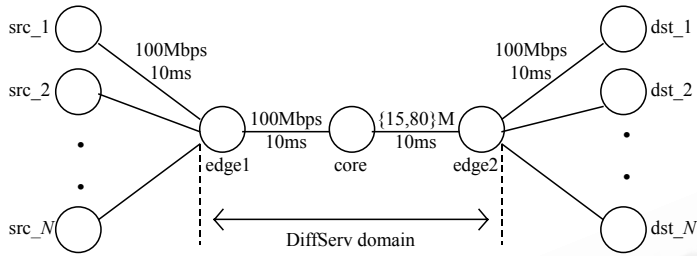


Fig.3 Network topology in simulation

图 3 仿真实验网络拓扑示意图

4.1 自适应RIO算法与RIO算法的稳定性比较

我们比较自适应RIO算法与原RIO算法在动态负载和不同订购比例条件下的时延和分组丢失性能的稳定性.实验配置如下:设置瓶颈链路带宽为 15Mbps,队列缓存区大小为 1500 个分组,TCP 分组平均大小为 1KB.设置 TCP 负载流数目随时间动态变化,其初始值为 50,从 0 时刻到 15s 内每隔 50s 增加 50 个流,从 200s~350s 内每隔 50s 减少 50 个流,部署 40 个短期随机 TCP 流作为背景 Web 流量.自适应 RIO 算法的调整周期为 0.5s,3 种算法的参数配置参见表 1.

Table 1 Parameter configurations in experiments on performance stability

表 1 稳定性实验中的参数配置

	RIO	ARIO-D	ARIO-L
$(th_{min}^{out}, th_{max}^{out}, p_{max}^{out})$	(100,600,0.10)	(100,600,0.10)	(100,600,0.10)
$(th_{min}^{in}, th_{max}^{in}, p_{max}^{in})$	(600,800,0.01)	(600,800,0.01)	(1200,1500,0.01)
Adaptation parameters	-- --	$\alpha=0.1, \beta=0.9$	$\alpha=\beta=30$
Control parameters	-- --	$\gamma=0.5, \eta=0.25$	γ given by (11), $\eta=0.005$

4.1.1 瞬态性能比较

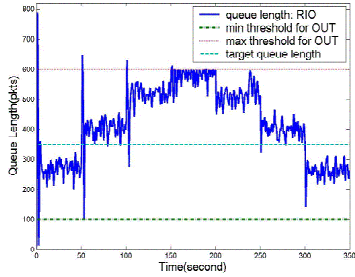
调整链路订购比例为不同的值,记录各算法的瞬时时延和分组丢失性能结果.因篇幅所限,在此仅给出链路订购比例为 60%时的实验结果,如图 4 中,(a)(b)表示 RIO 和 ARIO-D 算法的瞬时队列长度,(c)(d)表示 RIO 和 ARIO-L 算法的瞬时分组丢失率。

如图 4(a)、图 4(c)所示,RIO 的瞬时平均队列长度和分组丢失率随着 TCP 负载流量的变化而呈现大幅线性变化,符合前文中式(5)对 RIO 稳态性能分析的结论.如图 4(b)、图 4(d)所示,自适应 RIO 算法可以有效地减少时延或者分组丢失性能的瞬时波动,其中 ARIO-D 将平均队列长度有效地控制在目标区间之内(约 350 个分组),ARIO-L 算法将平均分组丢失率波动限制在控制目标(约 2%)的附近区间内.当链路的订购比例为其他值时,获得类似的实验结果。

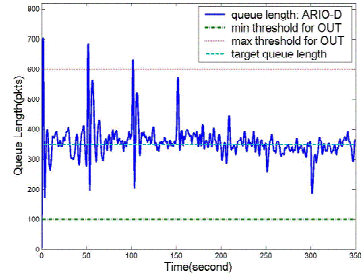
4.1.2 稳态性能比较

调整链路订购比例为不同的值,记录各算法在动态网络负载和不同订购比例下的稳态性能结果,作出随着订购量比例增加的性能曲线,如图 5 所示。

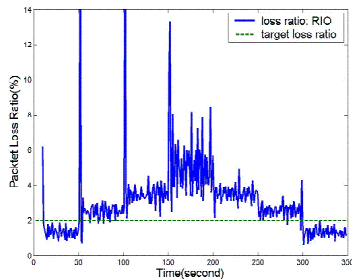
图 5(a)表示了 3 种算法的整体吞吐量和订购内吞吐量的实验结果,几乎重合的 3 种算法的性能曲线,表明 ARIO-D 算法和 ARIO-L 算法保持了 RIO 算法高带宽利用率和优先保证订购内流量的优点.图 5(b)表示出 3 种算法的平均队列占用率的实验结果,RIO 算法在不同订购量条件下的队列占用率的最大波动约 10%,而 ARIO-D 算法在各条件下都将队列占用率稳定在 45%左右.图 5(c)是 3 种算法的平均分组丢失率的实验结果,RIO 算法在不同订购量条件下的分组丢失率均高于 3%,而 ARIO-L 算法将各条件下的分组丢失率降低 1%.图 6(b)、图 6(c)也说明了两种自适应算法在队列时延和分组丢失率性能之间的折衷,当网络条件发生变化时,其中一种性能的提升是以另一种性能为代价的。



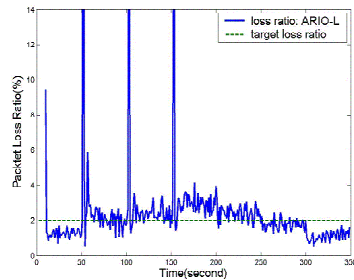
(a) Queue length: RIO, 60% subscription
(a) 队列长度:RIO 算法,订购比例为 60%



(b) Queue length: ARIO-D, 60% subscription
(b) 队列长度:ARIO-D 算法,订购比例为 60%



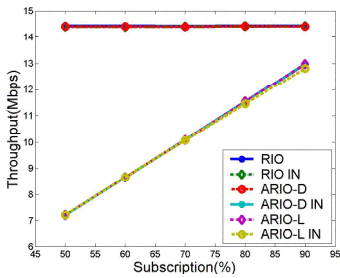
(c) Loss ratio: RIO, 60% subscription
(c) 分组丢失率:RIO 算法,订购比例为 60%



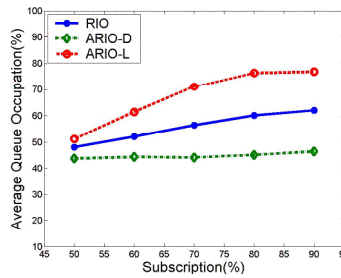
(d) Loss ratio: ARIO-L, 60% subscription
(d) 分组丢失率:ARIO-L 算法,订购比例为 60%

Fig.4 Instantaneous performance comparison among ARIO, ARIO-D and ARIO-L

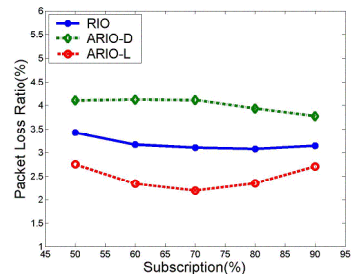
图 4 RIO 和 ARIO-D, ARIO-L 算法瞬态性能的比较



(a) Throughput
(a) 吞吐量



(b) Queue occupancy
(b) 队列占用率



(c) Loss ratio
(c) 分组丢失率

Fig.5 Steady performance comparison among RIO, ARIO-D and ARIO-L

图 5 RIO 和 ARIO-D, ARIO-L 算法稳态性能的比较

本组瞬态性能比较实验表明,ARIO-D和 ARIO-L 算法提高了原 RIO 算法的稳定性,可以在动态负载和不同订购比例下提供稳定的时延和分组丢失性能.

4.2 自适应RIO算法与原RIO算法的服务可区分能力比较

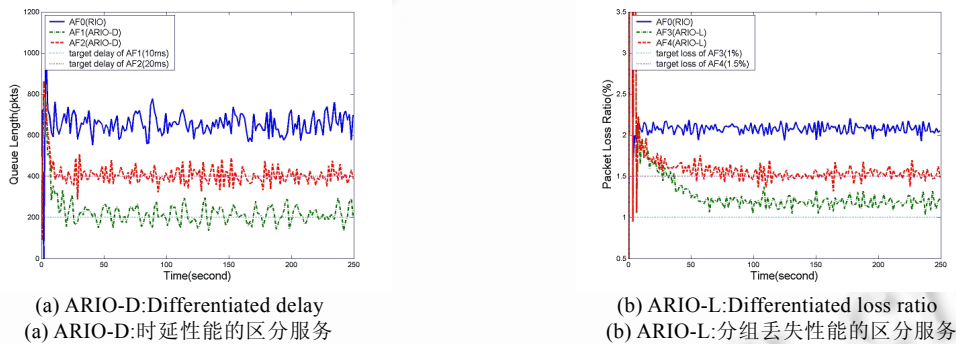
我们比较自适应 RIO 算法与原 RIO 算法在时延和分组丢失率方面的服务可区分的能力.实现一个采用 RIO 算法的单 AF 队列系统(AF0),一个采用 ARIO-D 和 ARIO-L 的 4 队列多服务系统(AF1-AF4).实验配置如下:瓶颈链路各服务队列的带宽为 80Mbps,队列缓存区大小为 6000 个分组,TCP 分组平均大小为 0.5KB,各服务队列的带宽订购比例为 60%;400 个长期 TCP 流构成主要负载,40 个短期随机 TCP 流作为背景流量.自适应 RIO 算法的调整周期为 0.5s,各服务队列的服务目标和算法配置参数,见表 2.

Table 2 Parameter configurations in experiments on differentiated service ability**表 2** 服务可区分能力实验中的参数配置

	AF0	AF1	AF2	AF3	AF4
PHB	IN throughput=48M	IN throughput=48M	IN throughput=48M	IN throughput=48M	IN throughput=48M
AQM	RIO	ARIO-D	ARIO-D	ARIO-L	ARIO-L
$(th_{min}^{out}, th_{max}^{out}, p_{max}^{out})$	(50,1250,0.10)	(50,1250,0.10)	(50,1250,0.10)	(50,1250,0.10)	(50,1250,0.10)
$(th_{min}^{in}, th_{max}^{in}, p_{max}^{in})$	(1250,1750,0.01)	(1250,1750,0.01)	(1250,1750,0.01)	(5500,6000,0.01)	(5500,6000,0.01)
Adaptation parameters	--	$\alpha=0.1, \beta=0.9$	$\alpha=0.1, \beta=0.9$	$\alpha=\beta=100$	$\alpha=\beta=100$
Control parameters	--	$\gamma=0.154, \eta=0.05$	$\gamma=0.308, \eta=0.05$	$\gamma=0.250, \eta=0.005$	$\gamma=0.375, \eta=0.005$

记录实验中 AF0~AF4 各队列的吞吐量、瞬时队列时延、瞬时分组丢失率等数据。AF0~AF4 队列的链路利用率和 IN 链路利用率都高达 91%~93%之间。图 6(a)示出 AF0 与 AF1~AF2 的瞬时队列曲线,图 6(b)示出 AF0 与 AF3~AF4 的瞬时分组丢失率曲线。在图 6(a)中,AF0 队列在 RIO 算法的作用下获得的瞬时时延约为 40ms(约 600pkts),而 AF1-2 队列在 ARIO-D 算法的作用下都基本达到 10ms(约 200 分组)和 20ms(约 400 分组)的时延区分目标。在图 6(b)中,AF0 队列在 RIO 算法的作用下获得的瞬时分组丢失率约为 2%,而 AF3~AF4 队列在 ARIO-L 算法的作用下基本达到提供区分的分组丢失率的目标,其中 AF3 队列的分组丢失率达到 1.5%左右的控制目标,AF4 队列获得了与控制目标 1%相近的 1.2%左右的性能。

本组服务可区分能力实验表明,ARIO-D 和 ARIO-L 可以提供定量的可区分的队列时延和分组丢失率性能。灵活部署和配置这两种算法可以提供吞吐量和时延或分组丢失的多种服务质量保证。

**Fig. 6** Ability of ARIO-D and ARIO-L to provide differentiated service**图 6** ARIO-D 和 ARIO-L 算法提供区分服务的能力

5 结束语

现有区分服务的保证转发可以提供带宽性能的保证,却缺乏对时延和分组丢失率性能保证的有效方案。主动队列管理算法是实现单节点路由器上的服务质量保证的主要技术。本文以在单节点上提供稳定定量的带宽、时延或者分组丢失率为研究目的,对保证转发中主要的主动队列管理算法 RIO 进行改进。仿真实验结果表明,本文提出的两种自适应 RIO 算法(ARIO-D 和 ARIO-L),不但保持了 RIO 算法对订购带宽的保证能力,而且具备提供队列时延和分组丢失服务质量保证的能力(即性能稳定性和服务可区分性)。这两种算法分别适用于时延敏感和分组丢失敏感的流量传输,基于它们实现的保证转发服务,不但可以简便地实现域内的时延和分组丢失率的定量性能保证,而且可以通过组合为多媒体流量提供灵活多样的 AF PHB 性能。

References:

- [1] Blake S, Black D, Carlson M, Davies E, Wang Z, Weiss W. An architecture for differentiated services. RFC 2475, 1998.
- [2] Nichols K, Carpenter B. Definition of differentiated services per domain behaviors and rules for their specification. RFC 3086, 2001.
- [3] Heinanen J, Baker F, Weiss W, Wroclawski J. Assured forwarding PHB group. RFC 2597, 1999.

- [4] Davie B, Charny A, Bennett JCR, Benson K, Boudec JYL, Courtney W, Davari S, Viroiu V, Stiliadis D. An expedited forwarding PHB (Per-Hop Behavior). RFC 3246, 2002.
- [5] Clark D, Fang W. Explicit allocation of best-effort packet delivery service. *IEEE/ACM Trans. on Networking*, 1998,6(4):362–373.
- [6] Dovrolis C. Proportional differentiated services for the internet [Ph.D. Thesis]. University of Wisconsin-Madison, 2000.
- [7] Christin N. Quantifiable service differentiation for packet networks [Ph.D. Thesis]. University of Virginia, 2003.
- [8] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. on Networking*, 1993, 1(4):397–413.
- [9] Chait Y, Hollot CV, Misra V, Towsley D, Zhang H, Lui JCS, Providing throughput differentiation for TCP flows using adaptive two-color marking and two-level AQM. In: *Proc. of the IEEE INFOCOM 2002*. Piscataway: IEEE Comput. Soc., 2002. 837–844.
- [10] Witosuraput S, Lambert J, Ouveysi I. Optimization flow control for aggregate traffic handling in the differentiated services internet. In: *Proc. of the IEEE ICC 2002*. Piscataway: IEEE Press, 2002. 965–969.
- [11] Mohamed AEG, Bose A, Shin KG. Evolution of the Internet QoS and support for soft real-time applications. *Proc. of The IEEE*, 2003,91(7):1086–1104.
- [12] Handley M, Floyd S, Padhye J, Widmer J. TCP friendly rate control (TFRC): Protocol Specification. RFC 3448, 2003.
- [13] Le CH, He JH, Liu W, Yang ZK, Performance of TFRC for multimedia services with bandwidth guarantee. In: *Proc. of the IEEE TENCON 2002*. Beijing: IEEE Press, 2002. 15–18.
- [14] Yeom I, Reddy ALN. Modeling TCP behavior in a differentiated services network. *IEEE/ACM Trans. on Networking*, 2001,9(1): 31–46.
- [15] Firoiu V, Borden M. A study of active queue management for congestion control. In: *Proc. of the IEEE INFOCOM 2000*. IEEE Press, 2000. 1435–1444.
- [16] Padhye J, Firoiu V, Towsley DF, Kurose JF. Modeling TCP Reno performance: A simple model and its empirical validation. *IEEE/ACM Trans. on Networking*, 2000,8(2):133–145.
- [17] Firoiu V, Yeom I, Zhang X. A framework for practical performance evaluation and traffic engineering in IP networks. In: *Proc. of the IEEE ICT 2001*, IEEE Press, 2001.
- [18] Feng W, Kandlur DD, Saha D, Shin KG. A self-configuring RED gateway. In: *Proc. of IEEE INFOCOM'99*. IEEE Press, 1999. 1320–1328.
- [19] Floyd S, Gummadi R, Shenker S. Adaptive RED: An algorithm for increasing the robustness of RED. Berkeley: The ICSI Center for Internet Research, 2001.
- [20] Shin J, Kim JW, Kuo CCJ. Quality-of-Service mapping mechanism for packet video in Differentiated Services network. *IEEE Trans. on Multimedia*, 2001,3(2):219–231.
- [21] The University of Southern California. Information Science Institute. The Network Simulator—ns-2. <http://www.isi.edu/nsnam/ns>