

基于参数不敏感设计的网络拥塞控制算法

刘 治¹, 倪 杰¹, 文俊朝¹, 章 云¹, 王耀南²

(1. 广东工业大学 自动化学院, 广东 广州 510640; 2. 湖南大学 电气与信息工程学院, 湖南 长沙 410006)

摘要: 本文提出了一种新颖的主动队列管理(active queue management, AQM)策略——间隔随机早期检测(interval random early detection, IRED). 与传统的RED机制不同, IRED的参数设计中, 平均队长的门限值-最小阈值和最大阈值从固定不变的单值, 变为了一个阈值区间. 相对于RED的单值固定阈值的设计, IRED的适应能力和鲁棒性得到增强, 在大多数的网络环境下能够保持良好的性能, 特别是在网络状态出现波动和突变时, 其性能比传统的AQM算法更佳. 本文进一步采用了TCP-AQM的动态模型分析了使用IRED控制器的拥塞控制系统的稳定性, 分析了系统的稳定裕度, 并给出了稳定裕度与控制增益的关系式, 从而提出了一种根据期望稳定裕度来设计阈值区间的系统化方法. 最后, 在NS-2仿真平台上, 通过与RED和Gentle-RED两种算法比较, 证明了IRED的优越性, 特别是在变负载的网络环境下的良好性能.

关键词: AQM; 拥塞控制; IRED; RED

中图分类号: TP273 **文献标识码:** A

Network congestion control algorithm based on parameter-insensitive design

LIU Zhi¹, NI Jie¹, WEN Jun-chao¹, ZHANG Yun¹, WANG Yao-nan²

(1. Department of Automation, Guangdong University of Technology, Guangzhou Guangdong 510006, China;

2. School of Electrical and Information Engineering, Hunan University, Changsha Hunan 410006, China)

Abstract: A novel active queue management(AQM) scheme called IRED(interval random early detection) is proposed to avoid the network congestion. Being different from the RED (random early detection) algorithm, the IRED uses the interval parameter mechanism to reduce the control sensitivity to the network variation, which improves the adaptive and robust performance of the network system. With the interval design, IRED achieves a better performance in the busy network environment. Furthermore, the closed-loop stability of the congestion control system is studied based on the transmission control protocol-active queue management(TCP-AQM) dynamic fluid model. By analyzing the stability margin of the congestion control system, we propose a systematic design method to design the interval parameter according to the expected stability performance. Finally, the simulation experiment is studied to verify the superior performance of IRED in a varying network environment.

Key words: AQM; congestion control; IRED; RED

1 引言(Introduction)

RFC2309 推荐 Floyd 等人提出的随机早期检测(random early detection, RED)作为AQM的唯一候选算法, 并得到路由器厂商, 诸如Cisco, Juniper等公司的广泛支持^[1]. RED通过检测路由器输出端口的平均队长 q_{avg} 来探测拥塞, 一旦 $q_{avg} > \min_{th}$ (最小阈值)就以一定概率标记或丢弃报文并通知源端拥塞状况. 与尾丢弃相比, RED算法引入了2个新机制: 1) 满队列前用概率机制提前丢弃分组, 解决满队列问题; 2) 用平均队长而不是即时队长来调整丢弃概

率, 尽可能吸收短暂突发流, 避免全局同步. RED的有效性经过了一系列实践的验证, 但没有完全解决全局同步, 而且当瞬时队长大于平均队长时, 一旦缓存耗尽就会工作在尾丢弃模式. RED的3大缺陷为^[2~5]: 1) 参数的敏感性, RED难以找到适应任何环境的参数, 参数的微小变化会给主体性能带来很大影响, 当连接数量或负载等网络环境变化时, 原本适应的参数将不再适应新环境, 反而带来不利影响; 2) 存在延时抖动, 重载时, 平均队长振荡在 $\max_{th}$ 附近, 产生延时抖动, 网络吞吐量急剧下降; 3) 无区分

收稿日期: 2008-05-30; 收修改稿日期: 2009-03-04.

基金项目: 国家-广东联合基金资助项目(U0735003); 国家自然科学基金资助项目(60604006, 60974047); 广东省自然科学基金资助项目(8351009001000002, 6021452).

服务,无法提供有效的、完全的公平性保障。

而今,围绕AQM和RED的研究十分丰富.针对RED的缺陷,已经出现了很多改进算法.WRED通过为不同的业务等级设定不同的最大丢弃概率 $p_{\max}$ 来提供不同的服务质量^[3].AdaptiveRED(ARED)将AQM的工作区间用阈值 $\min_{th}$ 和 $\max_{th}$ 划分为3部分^[5]: $q_{avg} < \min_{th}$ 为轻载区间,进入此区间上用 $p_{\max}/\alpha$ 修正 $p_{\max}$; $\min_{th} \leq q_{avg} \leq \max_{th}$ 为正常区间 $p_{\max}$ 不作修正; $q_{avg} > \max_{th}$ 为重载区间,进入此区间用 $p_{\max} \times \beta$ 修正 $p_{\max}$ ($\alpha > 1, \beta > 1$).FRED采用“每活跃流记账”(per-active-flow accounting)的方式来管理队列,将网络中的流大致分为:非适应流、脆弱流和强壮流,只向有较多报文在缓冲区排队的流发出拥塞通知,所以FRED可以起到公平分配带宽,保护脆弱流的作用^[3].但这些算法都是启发式的缺乏系统性.基于经典控制理论的控制模型的引入解决了队列管理算法缺乏系统性的问题^[6~10],比如V Misra等人在2000年基于流体理论提出的网络模型,忽略了TCP最大窗口限制、往返时延(RTT)差异等一系列环节,恰当描述了TCP传输流的行为,为研究人员广为采用.在这个模型的基础上,出现了对于RED算法的稳定性分析、PI/PID控制器、鲁棒控制器等AQM新技术的研究^[3].

考虑到RED对参数的敏感性,一组预设参数无法在变化的网络环境中取得良好的控制性能.当参数不是一个固定值而是在一定的区间内取值的话,RED的适应性能得到增强.以此思想为基础,本文提出了一种新颖的RED算法—Interval RED(IRED).IRED的参数设计中,平均队长的门限值—最小阈值和最大阈值从固定不变的单值,变为了一个阈值区间.相对于RED的单值固定阈值的设计,IRED缓减了RED参数过于敏感、难以配置的缺陷,其鲁棒性增强,本文在TCP动态流体模型下,系统地分析了控制系统的稳定性和稳定裕度,提出了使用期望的稳定裕度来配置系统工作区间的方法.

2 问题的提出(Problem formulation)

2.1 RED概述(Overview of RED)

RED通过指数加权滑动加权平均(EWMA)得到路由器缓存的平均队长(q_{avg}),根据 q_{avg} 得到临时丢弃概率 p_b ,见式(1);再由 p_b 得到最终丢弃概率 p_a ,见式(2):

$$p_b = \begin{cases} 0, & q_{avg} \leq \min_{th}, \\ \frac{p_{\max}(q_{avg} - \min_{th})}{\max_{th} - \min_{th}}, & \min_{th} < q_{avg} < \max_{th}, \\ 1, & q_{avg} \geq \max_{th}, \end{cases} \quad (1)$$

$$p_a = \frac{p_b}{1 - \text{count} \times p_b}. \quad (2)$$

其中: count表示连续成功传输的分组数, p_a 随count的增加而缓慢增加,保证了丢弃分组的均匀分布.

$\min_{th}$ 和 $\max_{th}$ 的设置对RED性能的影响举足轻重,它们的设置决定了RED控制的强弱,并影响控制的稳定性、快速性;此外, $\min_{th}$ 和允许的突发业务相关, $\min_{th}$ 越大,允许业务的突发性可越强; $\max_{th}$ 和最大时延相关, $\max_{th}$ 越大,排队延时越大.与不同的网络环境其相匹配的 $\min_{th}$ 和 $\max_{th}$ 是不一样的,RED中 $\min_{th}$ 和 $\max_{th}$ 的单一设定大大降低了其性能.

2.2 网络动态系统建模(Network dynamic system modeling)

2000年, V Misra等人提出了TCP/AQM的网络动态模型,该模型用式(3)和式(4)的非线性微分方程来描述TCP流行为.式(3)对TCP拥塞机制建模,描述了TCP的AIMD策略;式(4)对中间节点建模,描述了路由器缓存空间的变化:

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)W(t-R(t))p(t-R(t))}{2R(t-R(t))}, \quad (3)$$

$$\dot{q}(t) = \frac{W(t)N(t)}{R(t)} - C. \quad (4)$$

其中: W 为TCP期望窗口(单位:包); q 为期望队长(包); R 为往返时间(s); C 为链路容量(包/s); N 为负载因子,即TCP连接数; p 为丢弃率.

在系统的稳定工作点(W_o, q_o, p_o)附近进行局部线性化,可得到TCP-AQM反馈控制系统的框图,如图1所示.

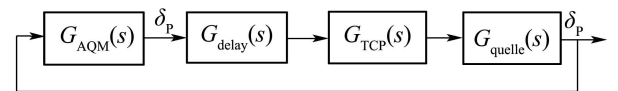


图1 TCP/AQM反馈控制系统模型框图

Fig. 1 TCP/AQM model of feedback control system block diagram

图中: $G_{AQM}(s)$ 是AQM控制器的传递函数; $G_{delay}(s)$ 是延迟环节的传递函数; $G_{tcp}(s)$ 是TCP的动态窗口的传递函数; $G_{queue}(s)$ 是动态队列的传递函数:

$$G_{delay}(s) = e^{-sR_o} G_{tcp}(s) =$$

$$\frac{\frac{R_o^2}{2N^2}}{s + \frac{R_o^2}{2N^2}} G_{\text{queue}}(s) = \frac{\frac{N}{R_o}}{s + \frac{1}{R_o}}. \quad (5)$$

系统的输入(平均队长)与输出(丢包率)的关系可由其控制对象的传递函数 $G_{\text{plant}}(s)$ 表征:

$$G_{\text{plant}}(s) = G_{\text{tcp}}(s) G_{\text{queue}}(s) G_{\text{delay}} = \frac{(R_o C)^3}{(2N)^2} e^{-sR_o} \cdot \frac{s}{(1 + \frac{s}{\frac{2N}{R_o^2 C}})(1 + \frac{s}{\frac{1}{R_o}})}. \quad (6)$$

则TCP/AQM系统的开环传递函数为

$$G(s) = G_{\text{AQM}}(s) G_{\text{plant}}(s) = \frac{(R_o C)^3}{(2N)^2} e^{-sR_o} \cdot G_{\text{AQM}}(s) \times \frac{s}{(1 + \frac{s}{\frac{2N}{R_o^2 C}})(1 + \frac{s}{\frac{1}{R_o}})}. \quad (7)$$

3 IRED算法(IRED algorithm)

考虑到网络环境的不确定性, 与其相适应的阈值 $\min_{\text{th}}$ 和 $\max_{\text{th}}$ 是不确定的, RED把阈值预设成了固定的单值, 这就大大影响了RED的性能. 如果分别给予 $\min_{\text{th}}$ 和 $\max_{\text{th}}$ 一个取值区间: $(\min_{\text{low}}, \min_{\text{high}})$ 和 $(\max_{\text{low}}, \max_{\text{high}})$, 必将能改善RED的鲁棒性, 得到一个RED的改进算法Interval RED(IRED), 如图2.

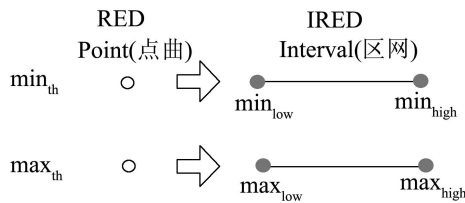


图 2 IRED的阈值设计

Fig. 2 IRED threshold design

只考虑阈值区间的上下边界 $\min_{\text{low}}, \min_{\text{high}}$ 和 $\max_{\text{low}}, \max_{\text{high}}$. 则IRED的设计步骤如下:

步骤 1 初始化阈值. 即设置阈值区间的边界值: $\min_{\text{low}}, \min_{\text{high}}$ 和 $\max_{\text{low}}, \max_{\text{high}}$. 在给出了边界值设置的系统方法.

步骤 2 计算 IRED 的丢包率. 对最小阈值($\min_{\text{low}}, \min_{\text{high}}$)和最大阈值($\max_{\text{low}}, \max_{\text{high}}$)进行组合, 得到4个阈值组合, 分别对应4个虚拟的RED控制器, 如表1. 其中 p_b 代表临时丢包率.

表 1 4个虚拟的RED控制器的临时丢包率和阈值

Table 1 Temporary packet loss rate and thresholds for four virtual RED controller

RED控制器	p_b	阈值
RED _{ll}	p_{ll}	$(\min_{\text{low}}, \max_{\text{low}})$
RED _{lh}	p_{lh}	$(\min_{\text{low}}, \max_{\text{high}})$
RED _{hl}	p_{hl}	$(\min_{\text{high}}, \max_{\text{low}})$
RED _{hh}	p_{hh}	$(\min_{\text{high}}, \max_{\text{high}})$

步骤 3 得到IRED的临时丢包率 p_{ired} . 由表1知道, 4个虚拟的RED控制器, RED_{ll}, RED_{lh}, RED_{hl}和RED_{hh}的临时丢包率分别为 p_{ll}, p_{lh}, p_{hl} 和 p_{hh} . 本文将IRED的临时丢包率 p_{ired} 设计为 p_{ll}, p_{lh}, p_{hl} 和 p_{hh} 的平均值(如表2), 得到如图3所示的IRED的临时丢包率和平均队长的关系图. 由图可见, 与RED的3个工作区不同, IRED 有5个工作区: 完全吸收区(absolute absorption interval, AAI)、最小阈值区(the $\min_{\text{th}}$ interval, MINI)、线性丢弃区(linear drop interval, LDI)、最大阈值区(the $\max_{\text{th}}$ interval, MAXI)和直接丢弃区(direct drop interval, DDI)分别对应于5类不同的控制对象, 即当平均队长在不同的范围时, IRED 会相应地工作在不同的工作区. 其中

$$f_{xy}(\min_x, \max_y) = \frac{p_{\text{max}}(q_{\text{avg}} - \min_x)}{(\max_y - \min_x)}.$$

表 2 IRED的工作区

Table 2 IRED working area

IRED的控制区间		临时丢包率				
名称	范围	p_{ll}	p_{lh}	p_{hl}	p_{hh}	p_{ired}
AAI	$q_{\text{avg}} < \min_{\text{low}}$	0	0	0	0	0
MINI	$\min_{\text{low}} < q_{\text{avg}} < \min_{\text{high}}$	f_{ll}	f_{lh}	—	—	$\frac{f_{ll} + f_{lh}}{4}$
LDI	$\min_{\text{high}} < q_{\text{avg}} < \max_{\text{low}}$	—	—	f_{hl}	f_{hh}	$\frac{f_{ll} + f_{lh} + f_{hl} + f_{hh}}{4}$
MAXI	$\max_{\text{low}} < q_{\text{avg}} < \max_{\text{high}}$	1	—	1	—	$\frac{f_{hh} + f_{lh} + 2}{4}$
DDI	$\max_{\text{high}} \sim \infty$	—	1	—	1	1

图3中的6个拐点分别为: $(\min_{\text{low}}, 0)$, $(\min_{\text{high}}, p_{\text{min}_{\text{th}}})$, $(\max_{\text{low}}, p_{\text{linear}})$, $(\max_{\text{low}}, p_{\text{max}_{\text{th}} - \text{low}})$, $(\max_{\text{high}}, p_{\text{max}_{\text{th}} - \text{high}})$ 和 $(\max_{\text{high}}, 1)$, 且有

$$p_{\text{min}_{\text{th}}} = \frac{p_{\text{max}}}{4} \left(\frac{\min_{\text{high}} - \min_{\text{low}}}{\max_{\text{low}} - \min_{\text{low}}} \right) + \frac{p_{\text{max}}}{4} \left(\frac{\min_{\text{high}} - \min_{\text{low}}}{\max_{\text{high}} - \min_{\text{low}}} \right), \quad (8)$$

$$p_{\text{linear}} = \frac{p_{\text{max}}}{4} \left(2 + \frac{\max_{\text{low}} - \min_{\text{low}}}{\max_{\text{high}} - \min_{\text{low}}} \right) + \frac{p_{\text{max}}}{4} \left(\frac{\max_{\text{low}} - \min_{\text{high}}}{\max_{\text{high}} - \min_{\text{high}}} \right), \quad (9)$$

$$p_{\text{max}_{\text{th}} - \text{low}} = \frac{1}{2} + \frac{p_{\text{max}}}{4} \left(\frac{\max_{\text{low}} - \min_{\text{low}}}{\max_{\text{high}} - \min_{\text{low}}} \right) + \frac{p_{\text{max}}}{4} \left(\frac{\max_{\text{low}} - \min_{\text{high}}}{\max_{\text{high}} - \min_{\text{high}}} \right), \quad (10)$$

$$p_{\text{max}_{\text{th}} - \text{high}} = \frac{1 + p_{\text{max}}}{2}. \quad (11)$$

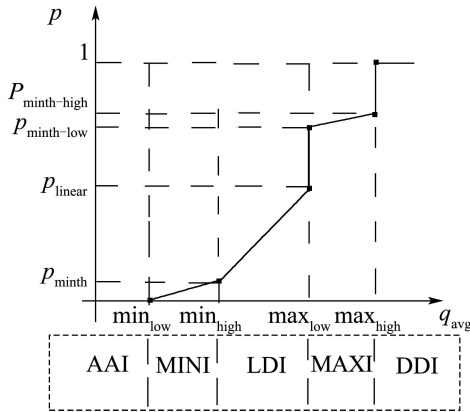


图3 IRED的临时丢包率和平均队长关系图
Fig. 3 Relationship of temporary packet loss rate and the average queue length for IRED

IRED控制器的传递函数为

$$G_{\text{ired}}(s) = \frac{L_{\text{ired}}}{1 + \frac{s}{k}}, \quad (12)$$

$$L_{\text{ired}} = \begin{cases} L_{\text{min}_{\text{th}}} = \frac{p_{\text{min}_{\text{th}}}}{\min_{\text{high}} - \min_{\text{low}}}, & q_{\text{avg}} \in \text{MINI}, \\ L_{\text{linear}} = \frac{p_{\text{linear}} - p_{\text{min}_{\text{th}}}}{\max_{\text{low}} - \min_{\text{high}}}, & q_{\text{avg}} \in \text{LDI}, \\ L_{\text{max}_{\text{th}}} = \frac{p_{\text{max}_{\text{th}} - \text{high}} - p_{\text{max}_{\text{th}} - \text{low}}}{\max_{\text{high}} - \max_{\text{low}}}, & q_{\text{avg}} \in \text{MAXI}. \end{cases} \quad (13)$$

其中: $k = \frac{\log_e(1 - w_q)}{\delta}$, δ 是采样频率.

4 IRED的稳定性和鲁棒性分析(Stability and robustness analysis of IRED)

4.1 IRED的稳定性定理(Stability theorem of IRED)

在下述定理1中, 通过对IRED的稳定性进行分析可知, IRED的系统稳定性可由适当的控制增益来保证.

定理1 在TCP/AQM反馈控制系统的动态模型中, 当 k 和 L_{ired} 满足(30)时, 所有采取IRED策略, 并满足 $N > N^-$, $R_o < R^+$ 的控制系统稳定.

$$\frac{L_{\text{linear}}(CR^+)^3}{(2N^-)^2} \leq \sqrt{1 + \left(\frac{\omega_g}{k}\right)^2}. \quad (14)$$

其中穿越频率 ω_g 为

$$\omega_g = \eta \min \left\{ \frac{2N^-}{(R^+)^2 C}, \frac{1}{R^+} \right\}. \quad (15)$$

证 略.

4.2 IRED的鲁棒性分析(Robustness analysis of IRED)

考虑到系统的稳定裕度可以指标其鲁棒性, 本文对TCP/IRED系统的鲁棒性分析立足于对其稳定裕度的分析上.

定理2 TCP/IRED系统中, 系统的稳定裕度受控制器的控制增益影响. 即

▲ 在工作区 MINI 上, 系统的相角稳定裕度 $\gamma_{\text{min}_{\text{th}}}$ 和幅值稳定裕度 $K_{\text{min}_{\text{th}}}$ 受IRED控制器在该工作区的控制增益 $L_{\text{min}_{\text{th}}}$ 的影响:

$$\gamma_{\text{min}_{\text{th}}} \geq 180^\circ - \arctan \sqrt{\left(\frac{L_{\text{min}_{\text{th}}}(R^+C)^3}{(2N^-)^2}\right)^2 - 1} - \eta \left(\frac{180}{\pi}\right)^\circ, \quad (16)$$

$$K_{\text{min}_{\text{th}}} \geq \frac{\frac{\pi}{2R^+}}{K \sqrt{\left(\frac{L_{\text{min}_{\text{th}}}(R^+C)^3}{(2N^-)^2}\right)^2 - 1}}. \quad (17)$$

▲ 在工作区 LDI 上, 系统的相角稳定裕度 γ_{linear} 和幅值稳定裕度 K_{linear} 受IRED控制器在该工作区的控制增益 L_{linear} 的影响:

$$\gamma_{\text{linear}} \geq 180^\circ - \arctan \sqrt{\left(\frac{L_{\text{linear}}(R^+C)^3}{(2N^-)^2}\right)^2 - 1} - \eta \left(\frac{180}{\pi}\right)^\circ, \quad (18)$$

$$K_{\text{linear}} \geq \frac{\frac{\pi}{2R^+}}{K \sqrt{\left(\frac{L_{\text{linear}}(R^+C)^3}{(2N^-)^2}\right)^2 - 1}}. \quad (19)$$

▲ 在工作区MAXI上, 系统的相角稳定裕度 γ_{maxth} 和幅值稳定裕度 L_{maxth} 受IRED控制器在该工作区的控制增益 L_{maxth} 的影响:

$$\begin{aligned} \gamma_{\text{maxth}} &\geq \\ 180^\circ - \arctan \sqrt{\left(\frac{L_{\text{maxth}}(R^+C)^3}{(2N^-)^2}\right)^2 - 1} - \\ \eta\left(\frac{180}{\pi}\right)^\circ, \end{aligned} \quad (20)$$

$$K_{\text{maxth}} \geq \frac{\frac{\pi}{2R^+}}{K \sqrt{\left(\frac{L_{\text{maxth}}(R^+C)^3}{(2N^-)^2}\right)^2 - 1}}. \quad (21)$$

证 略.

5 IRED的系统化设计(Systematic design of IRED)

IRED的系统设计中最重要的一环就是阈值区间的设计. 与RED在参数设计上根据经验的方法不同, 在IRED的阈值设计中引入了系统稳定裕度作为指导. 根据IRED不同工作区上期望的稳定裕度, IRED的阈值区间的设计有如下步骤:

步骤 1 根据IRED在MINI工作区上期望的稳定裕度 γ_{minth} 和 K_{minth} , 由式(22)(23)得到相应的 L_{minth} :

$$\begin{aligned} L_{\text{minth}} - \gamma &= \\ \frac{(2N^-)^2(\tan^2(180^\circ - \gamma_{\text{minth}} - \eta(\frac{180}{\pi})^\circ) + 1)}{(R^+C)^3}, \end{aligned} \quad (22)$$

$$L_{\text{minth}} - K = \frac{(2N^-)^2}{(R^+C)^3} \sqrt{\left(\frac{\pi}{2kR^+K_{\text{minth}}}\right)^2 + 1}, \quad (23)$$

$$L_{\text{minth}} = \min\{L_{\text{minth}} - K, L_{\text{minth}} - \gamma\}. \quad (24)$$

$L_{\text{minth}} - K$ 是对应于期望的幅值稳定裕度

K_{minth} 的控制增益 L_{minth} ; $L_{\text{minth}} - \gamma$ 是对应于期望的相角稳定裕度 γ_{minth} 的控制增益 L_{minth} .

步骤 2 根据 IRED 在LDI工作区上期望的稳定裕度 γ_{linear} 和 K_{linear} , 由式(25)(26)得到相应的 L_{linear} :

$$\begin{aligned} L_{\text{linear}} - \gamma &= \\ \frac{(2N^-)^2(\tan^2(180^\circ - \gamma_{\text{linear}} - \eta(\frac{180}{\pi})^\circ) + 1)}{(R^+C)^3}, \end{aligned} \quad (25)$$

$$L_{\text{linear}} - K = \frac{(2N^-)^2}{(R^+C)^3} \sqrt{\left(\frac{\pi}{2kR^+K_{\text{linear}}}\right)^2 + 1}, \quad (26)$$

$$L_{\text{linear}} = \min\{L_{\text{linear}} - K, L_{\text{linear}} - \gamma\}. \quad (27)$$

$L_{\text{linear}} - K$ 是对应于期望的幅值稳定裕度 K_{linear} 的控制增益 L_{linear} ; $L_{\text{linear}} - \gamma$ 是对应于期望的相角稳定裕度 γ_{linear} 的控制增益 L_{linear} .

步骤 3 根据IRED在MAXI工作区上期望的稳定裕度 γ_{linear} 和 K_{linear} , 由式(28)(29)得到相应的 L_{maxth} :

$$\begin{aligned} L_{\text{maxth}} - \gamma &= \\ \frac{(2N^-)^2(\tan^2(180^\circ - \gamma_{\text{maxth}} - \eta(\frac{180}{\pi})^\circ) + 1)}{(R^+C)^3}, \end{aligned} \quad (28)$$

$$L_{\text{maxth}} - K = \frac{(2N^-)^2}{(R^+C)^3} \sqrt{\left(\frac{\pi}{2kR^+K_{\text{maxth}}}\right)^2 + 1}, \quad (29)$$

$$L_{\text{maxth}} - K = \min\{L_{\text{maxth}} - K, L_{\text{maxth}} - \gamma\}. \quad (30)$$

$L_{\text{maxth}} - K$ 是对应于期望的幅值稳定裕度 K_{maxth} 的控制增益 L_{maxth} ; $L_{\text{maxth}} - \gamma$ 是对应于期望的相角稳定裕度 γ_{maxth} 的控制增益 L_{maxth} .

步骤 4 为工作区间MINI, LDI和MAXI选择合适的长度 l_{MIN} , l_{LDI} 和 l_{MAX} . 使得根据式(29)得到的控制增益 L_{minth} , L_{linear} 和 L_{maxth} 满足由式(22)和式(28)算出的值.

$$\begin{cases} L_{\text{minth}} = \frac{p_{\text{max}}}{4} \left(\frac{1}{l_{\text{MIN}} + l_{\text{LDI}}} + \frac{1}{l_{\text{MIN}} + l_{\text{LDI}} + l_{\text{MAX}}} \right), \\ L_{\text{linear}} = \frac{p_{\text{max}}}{2} \left(\frac{1}{l_{\text{LDI}}} + \frac{1}{2} \left(\frac{1}{l_{\text{MIN}} + l_{\text{LDI}} + l_{\text{MAX}}} + \frac{1}{l_{\text{MIN}} + l_{\text{MAX}}} + \frac{l_{\text{MIN}}}{(l_{\text{MIN}} + l_{\text{LDI}})l_{\text{LDI}}} \right) \right), \\ L_{\text{maxth}} = \frac{p_{\text{max}}}{2l_{\text{MAX}}} \left(1 - \frac{1}{2} \left(\frac{l_{\text{MIN}} + l_{\text{LDI}}}{l_{\text{MIN}} + l_{\text{LDI}} + l_{\text{MAX}}} + \frac{l_{\text{LDI}}}{l_{\text{LDI}} + l_{\text{MAX}}} \right) \right). \end{cases} \quad (31)$$

6 仿真研究(Simulation research)

下面进行一系列的试验来比较IRED与RED, GENTLE-RED三者间的性能差异, 本研究使用2.31版的NS-2平台进行仿真, 操作系统为UBUNTU7.04. 实验中采用 n 个FTP业务源和 m 个HTTP业务源来模拟实际应用中得网络负载情况,

所有业务源与路由器 r_1 之间的链路容量为1 Mbps, 延时为160 ms到240 ms之间的随机值; 瓶颈链路位于路由器 r_1 和 r_2 之间, 链路容量为15 Mbps, 延时40 ms; 除路由器 r_1 与 r_2 之间的链接以外, 其余链接均使用DropTail; 路由器 r_1 上的缓存大小设置为800个包(分组缺省大小为500 bytes), 各个业务源的分组大小也设置为500 bytes.

表3 不同负载情况中IRED, RED和GENTLE-RED的性能指标

Table 3 Performance indicators in the different load conditions of IRED, RED and GENTLE-RED

负载情况	性能指标	IRED	RED	GENTLE-RED
20 FTP 60 HTTP	吞吐量平均值	13.6956	13.6841	13.6949
	分组的丢弃率	0.001417	0.001452	0.001407
30 FTP 90 HTTP	吞吐量平均值	13.9217	13.9034	13.9112
	分组的丢弃率	0.003667	0.003691	0.003695
40 FTP 120 HTTP	吞吐量平均值	13.9038	13.8777	13.8969
	分组的丢弃率	0.004791	0.004815	0.004859
50 FTP 150 HTTP	吞吐量平均值	13.9413	13.9386	13.9292
	分组的丢弃率	0.005959	0.006000	0.006156
60 FTP 180 HTTP	吞吐量平均值	13.9867	13.9854	13.9854
	分组的丢弃率	0.007435	0.007454	0.007454
70 FTP 210 HTTP	吞吐量平均值	14.0185	14.0201	14.031
	分组的丢弃率	0.008790	0.008908	0.008985
80 FTP 240 HTTP	吞吐量平均值	14.0535	14.053	14.046
	分组的丢弃率	0.010390	0.010350	0.010265
90 FTP 270 HTTP	吞吐量平均值	14.0751	14.0751	14.0766
	分组的丢弃率	0.011821	0.011796	0.011682
100 FTP 300 HTTP	吞吐量平均值	14.081	14.081	14.081
	分组的丢弃率	0.013378	0.013335	0.013335
110 FTP 330 HTTP	吞吐量平均值	14.1035	14.1035	14.0949
	分组的丢弃率	0.015094	0.015013	0.014990
120 FTP 360 HTTP	吞吐量平均值	14.1106	14.1106	14.1106
	分组的丢弃率	0.016697	0.016704	0.016704
130 FTP 390 HTTP	吞吐量平均值	14.1258	14.1259	14.1259
	分组的丢弃率	0.018479	0.018511	0.018511
140 FTP 420 HTTP	吞吐量平均值	14.1333	14.1333	14.1333
	分组的丢弃率	0.019975	0.020044	0.020044
150 FTP 450 HTTP	吞吐量平均值	14.1462	14.1463	14.1463
	分组的丢弃率	0.021692	0.021748	0.021748
160 FTP 480 HTTP	吞吐量平均值	14.1541	14.154	14.154
	分组的丢弃率	0.023928	0.023713	0.023713
170 FTP 510 HTTP	吞吐量平均值	14.1694	14.1694	14.1694
	分组的丢弃率	0.025405	0.025409	0.025409
180 FTP 540 HTTP	吞吐量平均值	14.167	14.167	14.167
	分组的丢弃率	0.026786	0.027020	0.027020
17组负载情况的平均值	吞吐量平均值	14.0463	14.0429	14.0443
	分组的丢弃率	0.0138649	0.0138802	0.013881

为了说明 IRED 的参数设置对负载量的敏感度较低, 我们设计了以下一系列的实验. 首先设定好 4 种算法的各个参数, 在不改变算法参数的情况下, 改变负载量, 以此来观察 4 种算法在固定参数设置的情况下对不同负载量的敏感度.

▲ IRED 的参数设置为

$$\begin{aligned} \min_{\text{low}} &= 150, \quad \min_{\text{high}} = 155, \\ \max_{\text{low}} &= 700, \quad \max_{\text{high}} = 705, \\ p_{\max} &= 0.1. \end{aligned}$$

▲ RED 的参数设置为

$$\begin{aligned} \min_{\text{th}} &= 150, \quad \max_{\text{th}} = 700, \\ p_{\max} &= 0.1, \quad w_q = 0.00000133. \end{aligned}$$

▲ GENTLE_RED 的参数设置为

$$\begin{aligned} \min_{\text{th}} &= 150, \quad \max_{\text{th}} = 700, \\ p_{\max} &= 0.1, \quad w_q = 0.00000133. \end{aligned}$$

首先, 设置 20 个 FTP 业务源和 60 个 HTTP 业务源(分组大小设置为 500 bytes), 所有业务源从 0 s 开始工作, 直到实验结束为止, 而则实验时间设定为 200 s. 路由器 r_1 和 r_2 之间的链路分别采用 IRED, RED 和 GENTLE_RED 3 种算法, 进行 3 次实验, 记录并计算出各个算法下的吞吐量平均值和 200 s 实验过程中分组的丢弃率, 整理这些数据我们可以得出一组实验数. 为了比较各个算法在各种不同负载的情况下的各种性能指标, 在上面实验的基础上, 每增加 10 个 FTP 业务源和 30 个 HTTP 业务源就进行一次 4 种算法的比较实验, 一直把负载量增加到 180 个 FTP 业务源和 540 个 HTTP 业务源, 通过整理数据, 并计算各组数据的平均值, 可以得到如下表 3 的实验数据.

根据表 3 的数据, 详细比较每个实验中 IRED 算法, RED 和 GENTLE_RED 算法的吞吐量与分组丢弃率. 在负载量为 20 个 FTP 业务源、60 个 HTTP 业务源(以下只写出 FTP 业务源的数量)到 70 个 FTP 业务源这几组实验中, 使用 IRED 算法所获得的吞吐量较大. 在负载量为 120 个 FTP 业务源以上的几组实验中, 使用 IRED 算法和 RED, GENTLE_RED 算法所获得的吞吐量是基本相等的, 除了 FTP 业务源为 160 个的一组实验以外, 其他负载情况中, 使用 IRED 算法的分组丢弃率都比 RED 和 GENTLE_RED 算法的要小.

综上所述, 不难看出使用 IRED 算法在负载量较轻的时候(FTP 个数为 20~70)可以获得更大的带宽和更小的分组丢弃率. 当负载量增加到某个程度时, 如本组实验中负载量为 80 个 FTP 业务源、240 个 HTTP 业务源以上的时候, 两种算法所获得的吞吐量变得非常接近, 而重负载的时候(FTP 个数为 110~180)则是 IRED 算法的丢弃率较小. 由此可见, 在大多数的负载情况中, 使用 IRED 算法所获得的性能都比使用 RED 和 GENTLE_RED 的要好.

从平均值的结果, 进一步验证了使用 IRED 算法能够适应大部分的负载情况. 从表 4 中 17 组实验的平均数据来看, 采用 IRED 算法所获得的吞吐量平均值要比采用 RED 算法所获得的吞吐量要高, 与此同时采用 IRED 的分组丢弃率平均值却比采用 RED 和 GENTLE_RED 两种算法的分组丢弃率要低, 这就说明了在不同的各种负载环境中, 在算法参数不改变条件下, IRED 对负载的敏感度比 RED 和 GENTLE_RED 低. 在大部分的负载环境下, IRED 仍然能够获得较好的性能.

7 结论(Conclusion)

本文提出了一种新颖的 IRED 拥塞控制算法, 将传统单点阈值设计扩展为阈值区间设计. 相对于单值固定阈值的设计模式, IRED 改善了 RED 参数过于敏感、对负载状态敏感的缺陷, 在不牺牲算法复杂性和计算量的前提下增强了系统鲁棒性, 具有较强的实用性. 特别是在网络状态出现波动和突变时, 其性能比传统的 AQM 算法更佳. 本文结合网络流体动态模型系统分析了 IRED 的控制性能, 提出了一种根据期望稳定裕度来设计阈值区间的系统化方法. 进一步工作, 将研究以精简的学习修正模式改善 IRED 的自适应能力, 使其在更大范围内实现路由队列的平稳控制.

参考文献(References):

- [1] LIN D, MORRIS R. Dynamics of random early detection[C] // *Proceedings of ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*. New York: ACM Press, 1997: 127 – 138.
- [2] LIU S, BASAR T, SRIKANT R. Exponential-RED: a stabilizing AQM scheme for low-and high-speed TCP protocols[J]. *IEEE Transactions on Networking*, 2005, 13(5): 1068 – 1081.
- [3] RYU S, RUMP C, QIAO C. Advances in internet congestion control[J]. *IEEE Communications Surveys & Tutorials*, 2003, 5(1): 28 – 39.

- [4] OTT T J, LAKSHMAN T V, WONG L H. SRED: stabilized RED[C] // *Proceedings of the 18th Annual Joint Conference on the IEEE Computer and Communications Societies*. New York: IEEE, 1999: 1346 – 1355.
- [5] FLOYD S, GUMMADI R, SHENKER S. Adaptive RED: an algorithm for increasing the robustness of RED[DB/OL]. 2001. <http://www.icir.org/floyd/papers/adaptiveRed.pdf>.
- [6] RANJAN P, LA R, ABED E. Global stability conditions for rate control with arbitrary communication delays[J]. *IEEE Transactions on Networking*, 2006, 14(1): 94 – 107.
- [7] CHIU D M, JAIN R. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks[J]. *Computer Networks and ISDN Systems*, 1989, 17(1): 1 – 14.
- [8] KELLY F, MAULLOO A, TAN D. Rate control in communication networks: Shadow prices, proportional fairness and stability[J]. *Journal of the Operational Research Society*, 1998, 49(3): 237 – 252.
- [9] TIAN Y, YANG H. Stability of the internet congestion control with diverse delays[J]. *Automatica*, 2004, 40(9): 1533 – 1541.
- [10] TIAN Y. A general stability criterion for congestion control with diverse communication delays[J]. *Automatica*, 2005, 41(7): 1255 – 1262.

作者简介:

刘 治 (1977—), 男, 广东工业大学副教授, 博士, 主要研究方向为网络控制系统、智能控制系统, E-mail: lz@gdut.edu.cn;

倪 杰 (1983—), 女, 硕士研究生, 研究兴趣为网络控制、智能控制;

文俊朝 (1983—), 男, 硕士研究生, 研究兴趣为网络服务质量、智能控制;

章 云 (1963—), 男, 教授, 博士生导师, 研究兴趣为智能控制、网络化控制;

王耀南 (1957—), 男, 教授, 博士生导师, 研究兴趣为智能控制与信息处理、机器人.