

大时滞网络中的拥塞控制算法*

任丰原¹⁺, 林闯¹, 任勇², 山秀明²

¹(清华大学 计算机科学与技术系,北京 100084)

²(清华大学 电子工程系,北京 100084)

Congestion Control Algorithm in Large-Delay Networks

REN Feng-Yuan¹⁺, LIN Chuang¹, REN Yong², SHAN Xiu-Ming²

¹(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

²(Department of Electronic Engineering, Tsinghua University, Beijing 100084, China)

+ Corresponding author: Phn: 86-10-62783596, Fax: 86-10-62771138, E-mail: renfy@csnet1.cs.tsinghua.edu.cn

<http://www.cs.tsinghua.edu.cn>

Received 2002-01-22; Accepted 2002-07-01

Ren FY, Lin C, Ren Y, Shan XM. Congestion control algorithm in large-delay networks. *Journal of Software*, 2003,14(3):503~511.

Abstract: AQM (active queue management) can maintain the smaller queuing delay and higher throughput by the purposefully dropping the packets at the intermediate nodes. It is a hotspot in the current researches about TCP end-to-end congestion control. Almost all the existed algorithms neglect the impact on performance caused by large delay. In this study, firstly a fact through simulation experiments is verified, which is the queues controlled by several typical AQM algorithms, including RED, PI controller and REM, the dramatic oscillations in large delay networks are appeared, which decreases the utilization of the bottleneck link and introduces the avoidable delay jitter. After some appropriate model approximation, a robust AQM algorithm applying the principle of internal mode compensation in control theory is designed. The new algorithm restricts the negative impact on the queue stability caused by the large delay. The simulation experimental results show that the integrated performance of the proposed algorithm is obviously superior to that of the existed schemes when the network configuration parameters are large delay and small queue length, and the link utilization increases 3~4 times.

Key words: congestion control; internal mode control; flow control; large delay; active queue management

摘要: 主动队列管理(AQM)通过网络中间节点有目的的分组丢弃实现了较低的排队延时和较高的有效吞吐量,是近年来TCP端到端拥塞控制的一个研究热点.已有的大多数AQM算法在设计过程中都没有充分考虑到大

* Supported by the National Natural Science Foundation of China under Grant Nos.60273001, 60173012 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2001AA11208 (国家高技术研究发展计划); the National Grand Fundamental Research 973 Program of China under Grant No.G1999032707 (国家重点基础研究发展规划(973)); 985 Basic Research Foundation of the School of Information Science and Technology of Tsinghua University of China (清华大学信息学院 985 基础创新研究基金)

第一作者简介: 任丰原(1970-),男,甘肃临洮人,博士,讲师,主要研究领域为网络拥塞控制和流量控制,测控网络, Sensor network, Network calculus.

时滞对算法性能的影响.首先通过仿真试验证实了已有的几种典型算法控制的队列在大时滞网络中无一例外地出现了剧烈的振荡,导致瓶颈链路利用率下降和延时抖动加剧.为此,在进行了适当模型拟合处理的基础上,应用控制理论中的内模补偿原理设计了鲁棒的延时补偿主动队列管理(delay compensation-active queue management,简称 DC-AQM)算法,克服了大时滞给队列稳定性造成的不利影响.仿真实验结果表明,新算法在大时滞小期望队列长度的网络配置中表现出的综合性能明显优于已有的算法,链路利用率是其他算法的3~4倍.

关键词: 拥塞控制;内模控制;流量控制;大时滞;主动队列管理

中图法分类号: TP393 文献标识码: A

自1988年V.Jacobson提出了TCP端到端的、基于窗口的拥塞控制算法以来,TCP的流量控制算法已经过了Tahoe,Reno,New Reno^[1],SACK^[2]和Vegas^[3]等多个版本的增强与改进,所有这些工作都将研究的注意力集中到了终端系统上.近来的研究越来越强调网络中间节点在拥塞控制中的作用.显式拥塞标识(explicit congestion notification,简称ECN)对网络中间节点的功能进行了必要的扩展^[4],当路由器探测到即将到来的拥塞后,立即显式通告业务源采取适当的拥塞避免措施,这与依据连续接收 N 个重复ACK消息或超时计时器状态的隐式探测机制相比,显然要具有良好的响应性.最初提出随机早期探测(random early detection,简称RED)^[5]算法的目的是解决Drop Tail等队列管理算法可能造成TCP连接的全局同步、队列长时间处于满状态和处理突发业务时存在的公平性等问题.1998年,B.Braden等人提出了主动队列管理(active queue management,简称AQM)的研究倡议^[6],作为端到端拥塞控制的一种技术手段,期望AQM在减小排队时延的同时保证较高的吞吐量.RED符合AQM的技术目标,RFC2309推荐它为主动队列管理的惟一候选算法.接下来的研究和实践表明,RED在鲁棒性方面存在一定问题,它的性能敏感于设计参数和网络状况,在特定的网络负载状况下依然会导致多个TCP的同步,造成队列振荡、吞吐量降低和时延抖动加剧^[7].鲁棒的AQM策略成为目前Internet研究领域的技术热点,相继产生了不少有影响力的算法,如SRED^[8],BLUE^[9],PI控制器^[10],REM^[11],FRED^[12],Self-configuration RED^[13],Balanced-RED^[14]和虚队列AQM策略^[15-17].已有的大多数策略和算法都没有充分考虑往返时延(RTT)对算法性能的影响,在小时延的LAN或MAN上适用的算法未必能有效地工作在大时滞的WAN环境下.在本文的研究中,我们将着重考虑大时滞对AQM策略稳定性的影响,并采用控制理论中的内模补偿原理设计用于AQM的时延补偿控制器,以抵消时延对性能造成的负面影响.

1 大时延对AQM算法性能的影响

因为主机终端和网络中间节点在地理空间上的散落性,路由器在拥塞控制中产生的任何激励以及源终端系统作出的响应都有一定的滞后.在PI^[10]和REM^[11]的设计中,这一点均被忽略了.对于几毫秒到几十毫秒往返时延的LAN或MAN,这种忽略有其合理的因素,但在跨洲越洋的TCP连接中,RTT时间往往有几百毫秒,甚至达到秒的数量级.为了证实这一点,我们利用Ping程序从位于清华大学校内的166.111.64.239测试了去往不同地址的RTT时间,平均20个测试分组的响应结果见表1.在同一个LAN中,RTT只有11.3ms;但去往美国东海岸时RTT却高达886.7ms.有关RTT分布更为详尽的研究见文献[18].

Table 1 RTT statistical values

表1 RTT统计结果

| URL | Delay (ms) | URL | Delay (ms) |
|------------------------------|------------|------------------------------------|------------|
| www.edu.cn (202.112.0.36) | 11.3 | www.yahoo.com (216.115.102.79) | 395.1 |
| www.263.net (211.100.31.131) | 31.2 | www.whitehouse.org (64.226.222.72) | 886.7 |

已有的大多数关于AQM研究的文献,在验证算法性能时,无一例外地认定RTT时间不超过几十毫秒,仅具有LAN的拓扑特性.为了解几种典型算法在大时滞网络环境中的性能,我们在NS2^[19]平台上采用如图1所示的网络拓扑结构进行仿真实验.瓶颈链路位于节点A和节点B之间,链路容量15Mbps(3750packets/s,分组缺省大小为500bytes),延时5ms;所有业务源均为持久性的FTP业务源,它们与节点A之间的链路容量均为10Mbps,延时5ms.除节点A的队列由各种AQM算法控制以外,其余队列均为Drop Tail,所有节点的缓存大小均为

300packets,节点 B 和节点 C 之间的时延为 d ms.给定 $d=30$ 和 $d=190$ 分别作为 LAN 和 WAN 的代表.在仿真中分别取节点 A 的队列管理算法为 RED,PI,PID^[20]和 REM.在 RED 中,高低门限值分别为 100packets 和 200packets,设定其他各算法中队列长度的期望值为 150packets.图 2~图 5 给出了仿真结果,显然,在大时滞网络中,所有典型算法控制的队列均出现了大幅度振荡.队列的大幅度振荡一方面增加了端到端的时延抖动;同时由于空队列出现概率的加大,链路利用率的降低是肯定的,这两点都违背了最初提出的 AQM 设计目标^[6].表 2 总结了各种算法控制队列的统计特性和性能指标.分析表 2 的数据,时延加大后,瓶颈链路 AB 的利用率普遍下降近 10%,虽然 PI 依旧有较高的利用率,但观察图 3,队列明显不稳定,RED 和 PI 控制队列的标准偏差增加了 40,而 REM 和 PID 的情形更差,接近前两者的 2 倍,达到了 75,由此引入的端到端时延抖动对实现特殊业务的 QoS 保证是极为不利的.

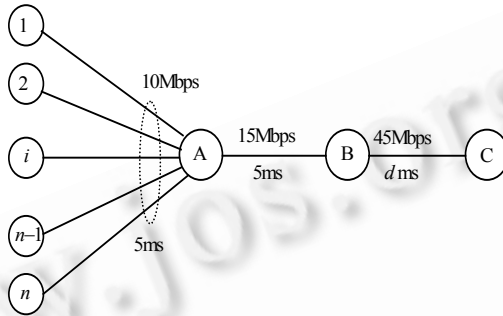


Fig.1 Simulation network topology

图 1 仿真网络拓扑结构

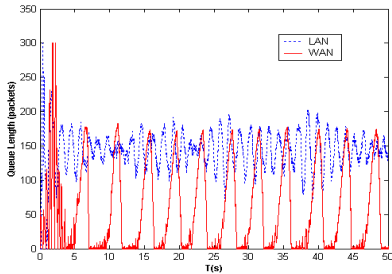


Fig.2 RED

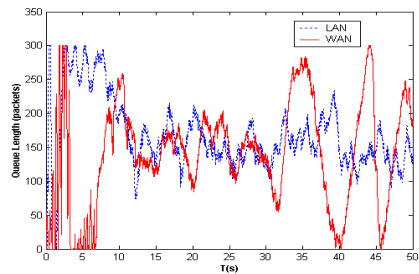


Fig.3 PI

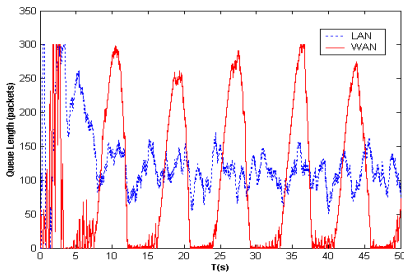


Fig.4 REM

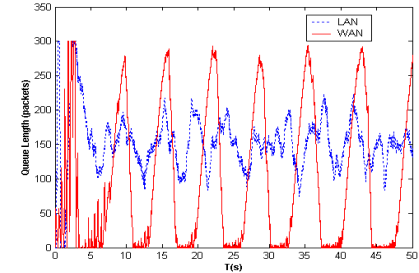


Fig.5 PID

Table 2 Queue length statistical values

表 2 队列长度的统计值

| Algorithm | RED | | PI | | REM | | PID | |
|----------------------|--------|-------|--------|--------|--------|--------|--------|-------|
| | LAN | WAN | LAN | WAN | LAN | WAN | LAN | WAN |
| Average queue length | 142.95 | 72.98 | 146.52 | 147.28 | 109.51 | 113.94 | 148.74 | 126.1 |
| Standard deviation | 19.89 | 61.15 | 20.95 | 64.62 | 23.57 | 98.01 | 29.41 | 98.75 |
| Link utilization (%) | 99.9 | 91.6 | 99.4 | 99.3 | 99.3 | 90.1 | 99.3 | 90.6 |

2 流量控制模型及其一阶近似

2.1 TCP流量控制中AQM系统的模型

在文献[21]中,V.Misra 等人基于流体流(fluid flow)理论建立了 AQM 作用下 TCP 连接上拥塞窗口的动态模型,它可以用下面的一组非线性微分方程来描述:

$$\begin{cases} \frac{dW(t)}{dt} = \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t)} p(t-R(t)) \\ \frac{dq(t)}{dt} = \frac{N(t)}{R(t)} W(t) - C(t) \end{cases} \quad (1)$$

为了便于分析系统的稳定性,C.Hollot 等人用小信号理论在稳态工作点对其进行了局部线性化^[22],得到了如图 6 所示的时滞二阶模型,图中控制器即为具体的 AQM 算法。

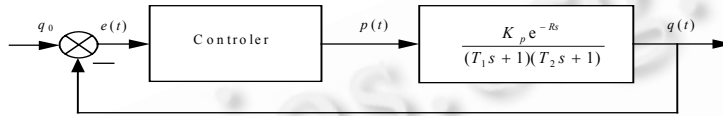


Fig.6 Block diagram of AQM system

图 6 AQM 控制框图

各参数的物理意义为 $K_p = \frac{(RC)^3}{4N^2}$, $T_1 = R$, $T_2 = \frac{R^2 C}{2N}$, 其中 C 为链路的容量, N 为激活 TCP 连接数, R 是往返时延。

2.2 时滞一阶模型拟合

在工业控制领域,许多被控对象的模型都可以用时滞一阶系统来很好地近似.对于这种模型,控制理论有许多成熟的方法来设计性能良好的控制器,为此,首先需要用一阶时滞模型来拟合图 6 中所描述的二阶时滞系统,也就是说寻找式(2)和式(3)等价时,两个系统之间的参数关系。

$$G_p(s) = \frac{K_p e^{-Rs}}{(T_1 s + 1)(T_2 s + 1)}, \quad (2)$$

$$\hat{G}_p(s) = \frac{K e^{-Ls}}{Ts + 1}. \quad (3)$$

经推导,式(2)和式(3)中参数间的关系如下:

$$\hat{G}_p(s) \approx \frac{K_p e^{(-T_1 - T_2 - R + \sqrt{T_1^2 + T_2^2})s}}{\sqrt{T_1^2 + T_2^2} s + 1}. \quad (4)$$

作为验证,取网络状态参数的典型值, $N=60$, $C=3750$ packets/sec, $R=0.4$, 我们有

$$G_p(s) = \frac{2.34 \times 10^5 \times e^{-0.4s}}{(0.4s + 1)(5.0s + 1)},$$

$$\hat{G}_p(s) = \frac{2.34 \times 10^5 \times e^{-0.784s}}{5.016s + 1}.$$

3 时延补偿的算法设计

3.1 内模控制

大时滞系统的控制问题一直是控制理论着力解决的一个主要问题.Smith 预估器是一种广泛应用的时延补偿方案,但它有一个致命的弱点,即对模型的误差非常敏感.作为对 Smith 预估器的一种扩展,20 世纪 80 年代初 Garcia 和 Morari 提出的内模控制(internal mode control,简称 IMC)^[23]在保持了控制器结构简单,设计直观、简便

等特点的基础上,克服了 Smith 预估器必须依赖精确模型的弱点,使系统的鲁棒性和抗扰性大为改善,而具有 PID 结构的内模控制器的一般设计方法由 Rivera^[24]等人给出.作为一个直接的方法,控制器的设计是基于一个假设的过程模型和一个用于鲁棒特性的低通滤波器,其方法是使用直接的两步 IMC 控制器设计,以获得一个具有传统 PID 结构的控制器.内模控制器的设计过程分为两步:

第 1 步.将模型分解成全通部分 G_{m+} 和最小相位部分 G_{m-} ,且

$$G_m = G_{m+} G_{m-},$$

使 G_{m+} 包括所有的 RHP 零点和时滞部分,即

$$|G_{m+}(j\omega)| = 1 \quad \forall \omega$$

依赖于目标函数的不同, G_{m+} 可以有不同形式的分解,如果采用绝对积分误差 (IAE),那么, G_{m+} 具有如下形式:

$$G_{m+} = e^{-\theta s} \prod_i (-\beta_i s + 1), \quad \text{Re}(\beta_i), \theta > 0.$$

如果采用平方积分误差 (ISE), G_{m+} 应该具有如下形式:

$$G_{m+} = e^{-\theta s} \prod_i \frac{-s + \beta_i}{s + \beta_i^H}, \quad \text{Re}(\xi_i), \theta > 0,$$

其中上标 H 表示复共轭.

第 2 步.鲁棒稳定性和鲁棒品质.

定义内模控制器:

$$G_{imc} = G_{m-}^{-1} f,$$

其中 f 是用户指定的低通滤波器,通常它的形式是

$$f = \frac{1}{(\lambda s + 1)},$$

由于反馈控制器 G_c 和内模控制器 G_{imc} 存在如下关系:

$$G_c = \frac{G_{imc}}{1 - G_{imc} G_m}, \quad (5)$$

因此,由内模控制设计的反馈控制器形式为

$$G_c = \frac{G_m^{-1} f}{1 - f G_{m+}}. \quad (6)$$

3.2 时延补偿的 AQM 算法 (DC-AQM)

在第 1 节中,我们仔细分析和评价了大时延对 AQM 算法稳定性的影响,认识到目前已有的大多数算法都没有很好地考虑网络中可能存在的大时滞现象,以致于在该网络环境下工作时,算法的性能急剧下降,无法实现 AQM 的技术目标.下面,我们利用内模控制为大时滞网络中的主动队列管理设计一种鲁棒的时延补偿控制算法,期望在保持队列稳定工作的同时,实现较高的利用率.

在第 2.2 节中,我们通过模型拟合得到了 TCP 流量控制的时滞一阶近似模型,用它作为被控制系统的模型:

$$G_m(s) = \frac{K e^{-Ls}}{Ts + 1}.$$

为了使所得的控制器具有简单、易实现的结构,我们用对称一阶 Pade 近似地来表示模型中的时滞环节:

$$G_m(s) \approx \frac{K \left(1 - \frac{L}{2}s\right)}{(Ts + 1) \left(1 + \frac{L}{2}s\right)}.$$

用绝对积分误差 (IAE) 最优分解,得到

$$G_{m-}(s) = \frac{K}{(Ts + 1) \left(1 + \frac{L}{2}s\right)}, \quad G_{m+}(s) = \left(1 - \frac{L}{2}s\right).$$

低通滤波器采用常用形式,即

$$f(s) = \frac{1}{(\lambda s + 1)}.$$

那么,相应的内模控制器为

$$G_{imc}(s) = \frac{(Ts + 1) \left(\frac{L}{2}s + 1 \right)}{K(\lambda s + 1)}.$$

文献[24]讨论了参数 λ 对控制器性能的影响,当 $\lambda/L \approx 0.8$ 时,系统具有良好的瞬态和稳态响应,这里我们沿用这一参数关系.

依据式(6),解得对应的反馈控制器,它具有经典 PID 控制器的结构,

$$G_c(s) = \frac{P(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s,$$

其中 $K_p = \frac{(2T + L)}{2.6KL}$, $K_i = \frac{1}{1.3KL}$, $K_d = \frac{T}{2.6K}$.

至此,我们得到了具有时滞补偿能力的 AQM 控制器.为方便起见,将该算法称为 DC-AQM 算法.在实际应用中,采样系统需要控制器的离散形式.以一系列的采样时刻点 kT 代表连续时间 t ,以和式代替积分,以增量代替微分,得到离散的 PID 表达式为

$$p(k) = K_p e(k) + K_i \sum_{j=0}^k e(j) + K_d [e(k) - e(k-1)], \quad (7)$$

表示成增量形式:

$$\Delta p(k) = K_p \left\{ \left(1 + \frac{T}{T_i} + \frac{T_d}{T} \right) e(k) - \left(1 + \frac{2T_d}{T} \right) e(k-1) + \frac{T_d}{T} e(k-2) \right\}, \quad (8)$$

其中 $T_i = \frac{K_p}{K_i}$, $T_d = \frac{K_d}{K_p}$, T 为采样时间.

假定瓶颈链路的容量为 15Mbps(3750packets/s),激活的 TCP 连接数为 60,典型的 RTT 时间为 0.4 秒,则模拟的 DC-AQM 控制器为

$$G_c(s) = 2.268 \times 10^{-5} + 4.193 \times 10^{-6}/s + 8.245 \times 10^{-6}s. \quad (9)$$

为方便比较,这里依旧采用文献[10]中给定的采样频率,即 160Hz,相应的采样时间 T_s 为 0.00625 秒,用式(8)得到控制器的离散形式:

$$c(k+1) = c(k) + 1.34183 \times 10^{-3} e(k) - 2.66093 \times 10^{-3} e(k-1) + 1.31913 \times 10^{-3} e(k-2). \quad (10)$$

这里, $e(k) = q(k) - q_0$, $c(k)$ 为控制器的输出, $q(k)$ 是第 k 时刻的队列长度采样值, q_0 为期望队列长度.考虑到实际问题的物理意义,将分组丢弃概率定义为

$$p(k) = \begin{cases} 0, & c(k) < 0 \\ c(k), & 0 \leq c(k) \leq 1. \\ 1, & c(k) > 1 \end{cases} \quad (11)$$

4 DC-AQM 算法的仿真与性能评价

我们在 NS2 平台上实现了 DC-AQM 算法,并用如图 1 所示的拓扑结构进行仿真实验来评价算法的性能.给定节点 B 和节点 C 之间的广播时延为 190ms,那么 RTT 时间至少为 400ms,期望的队列长度设定为 150packets.图 7 和图 8 分别描述了队列和分组丢弃概率的变化.不难看出,对于相同的网络参数配置,图 2~图 5 中队列的剧烈振荡在图 7 中已不复存在,这说明 DC-AQM 算法对网络中的大时滞进行了有效的补偿.

为了了解和比较期望的队列长度对链路利用率的影响,我们将最小的 RTT 时间固定在 400ms,期望队列长度从 5packets 变化到 150packets,将 DC-AQM 和其他典型 AQM 算法在链路 AB 上的利用率描绘在图 9 中.很明

显,DC-AQM 中期望队列长度对链路利用率的影响最小,REM 在较小的期望队列长度处仅仅只能达到 30%~40%的利用率,其他算法也或多或少地受期望队列长度的影响。

下面,将期望队列长度固定在 150packets,RTT 时间中的广播延时部分从 0ms 变化到 1 000ms,得到各算法在瓶颈链路上的利用率变化曲线如图 10 所示,虽然时延对所有的算法都有影响,但 DC-AQM 算法始终保持着最高的链路利用率,REM 受时延的影响最大,RED 和 PI 介于 REM 和 DC-AQM 之间,在 700ms 以前,PI 比 RED 有更高的链路利用率.最后,为了综合分析时延和期望队列长度对 AQM 算法的影响,我们将 RTT 时间中 0ms~1 000ms 的固定广播延时等分为 25 段,5packets~150packets 的期望队列长度以间隔 5 等分.在 XY 平面上形成网格.以网格顶点处的参数配置图 1 中的仿真网络,记录每次仿真中瓶颈链路的利用率,将结果用曲面形式表述在图 11~图 14 中,相比较而言,DC-AQM 算法有较明显的优势,即便是在最极端的参数配置下,依旧保持着近 85%的利用率,而 RED,PI 和 REM 分别只能达到约 75%,30%和 20%.比较图 11~图 14,不难发现,RED 和 DC-AQM 受期望队列长度的影响较小,在相同的 RTT 时间下,链路的利用率大致是相同的.对于 PI 和 REM 则不然,在小队列长度和大时滞网络中,REM 和 PI 控制的队列使瓶颈链路的利用率急剧下降,两组曲面不同的变化趋势清晰地说明了这一点.

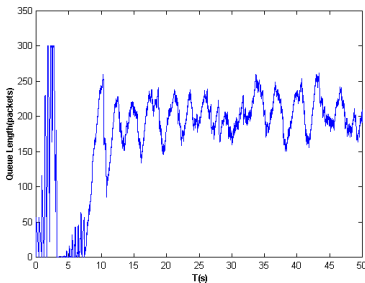


Fig.7 Log magnitude plots
图 7 队列长度

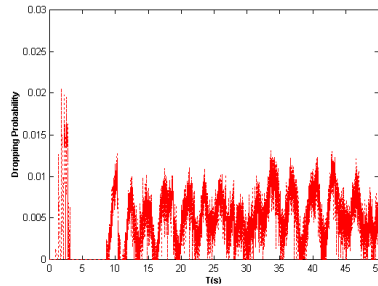


Fig.8 Log phase plots
图 8 分组丢弃概率

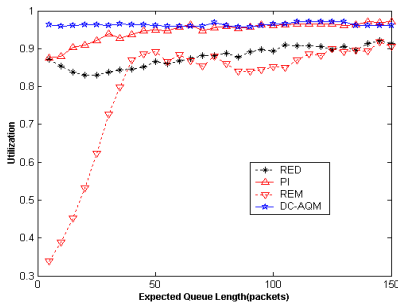


Fig.9 Relationship between IMC controller and feedback controller
图 9 利用率随队长的变化

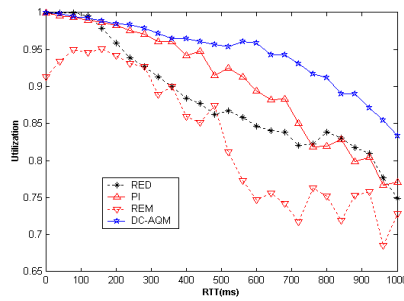


Fig.10 Queue evolution
图 10 利用率随时延的变化

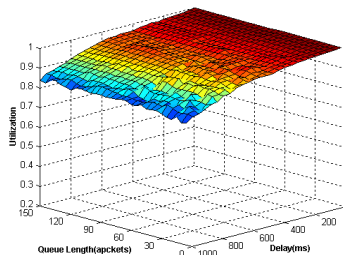


Fig.11 DC-AQM

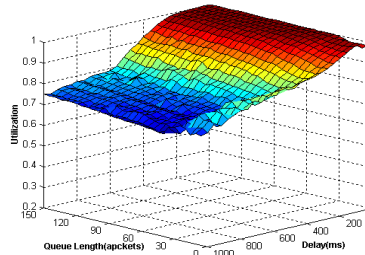


Fig.12 RED

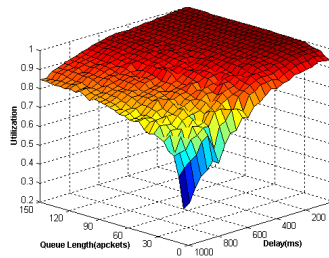


Fig.13 PI

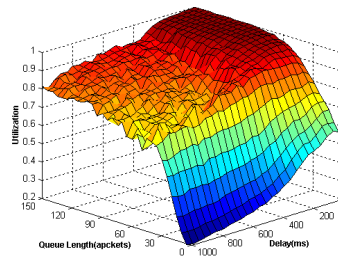


Fig.14 REM

5 结 论

主动队列管理是近年来 TCP 端到端拥塞控制研究的一个技术热点,它通过网络中间节点有控制的分组丢弃机制实现了较低的排队延时和较高的有效吞吐量.已有的大多数 AQM 算法在设计过程中都没有充分考虑到大时滞对算法性能的影响.在本文中,我们首先通过仿真实验研究了包括 RED、PI 和 REM 等几种典型算法在 RTT 时间较大的广域网中的性能,结果表明,它们控制的队列无一例外地出现了剧烈的振荡,导致瓶颈链路利用率下降,排队时延引入的延时抖动加剧.为此,我们在进行了适当模型拟合处理的基础上,应用控制理论中的内模补偿原理设计了鲁棒的延时补偿主动队列管理算法.仿真结果表明,算法大大抑制了队列的不稳定振荡,提高了链路利用率.此外,我们还通过仿真实验细致地分析和比较了 DC-AQM 算法和已有典型算法在不同 RTT 时间和不同期望队列长度下的性能.虽然在较低的 RTT 时间和较高的期望队列长度下,所有算法都表现出相当不错的性能,但在大延时小期望队列长度的情形下,DC-AQM 算法明显优于其他算法.当然,本文的研究仅仅局限在指出了大时滞对 AQM 算法性能的影响,所提出的算法虽然能够在大时延网络中有效地工作,但实际的网络系统是非常复杂的,不同的连接有着不同的 RTT 时间,即便可以等价出一个主导的 RTT 时间,它也必将动态变化的.如何综合应用辨识和自适应技术,设计出动态的 AQM 时延补偿算法应该是下一步工作的主要内容.

References:

- [1] Stevens W. TCP slow start, congestion avoidance, fast retransmit, and fast recovery. RFC 2001. 2001. <http://www.rfc.net/rfc2001.html>.
- [2] Fall K, Floyd S. Simulation-Based comparisons of Tahoe, Reno, and SACK TCP. Computer Communication Review, 1996,26 (3): 5~21.
- [3] Brakmo L, Peterson L. TCP vegas: end-to-end congestion avoidance on a global Internet. IEEE Journal on Selected Areas in Communication, 1995,13(8):1465~1480.
- [4] Floyd S. The addition of explicit congestion notification (ECN) to IP. 1996. <http://www.aciri.org/floyd/papers.html>.
- [5] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance. IEE/ACM Transactions on Networking, 1993,1(4):397~413.
- [6] Braden B, Braden B, Clark D, Crowcroft J, Davie B, Deering S. Recommendations on queue management and congestion avoidance in the Internet. RFC2309, 1998. <http://www.rfc.net/rfc2309.html>.
- [7] Floyd S. A report on some recent development in TCP congestion control. 2001. <http://www.aciri.org/floyd/papers.html>.
- [8] Ott TJ, Lakshman TV, Wong LH. SRED: stabilized RED. In: Proceedings of the INFOCOM'99. New York: IEEE Computer Society, 1999. 1346~1355.
- [9] Feng W, Kauldlur D, Saha D, Shin K. Blue: a new class of active queue management algorithms. Technical Report CSE-TR-387-99, University of Michigan, 1999.
- [10] Hollot C, Misra V, Towsley D. On designing improved controllers for AQM routers supporting TCP flows. In: Proceedings of the INFOCOM 2001. Alaska: IEEE Computer Society, 2001. 1726~1734.
- [11] Sanjeeva A, Steven HL. REM: active queue management. IEEE Network, 2001,15(3):48~53.

- [12] Lin D, Morris R. Dynamics of random early detection. In: Proceedings of the SIGCOMM'97. Cannes, 1997. 127~138.
- [13] Feng W, Kandlur D, Saha D, Shin K. A self-configuration RED gateway. In: Proceedings of the INFOCOM'99. New York: IEEE Computer Society, 1999. 1320~1328.
- [14] Anjum F, Tassiulas L. Balanced-RED: an algorithm to achieve fairness in Internet. 1997. <http://www.isr.umd.edu/CSHCN/>.
- [15] Kunniyur S, Srikant R. End-to-End congestion: utility function, random loss and ECN marks. In: Proceedings of the INFOCOMM 2000. Tel Aviv, 2000.
- [16] Kunniyur S, Srikant R. A time-scale decomposition approach to adaptive ECN marking. In: Proceedings of the INFOCOMM 2001. Alaska: IEEE Computer Society, 2001.
- [17] Kunniyur S, Srikant R. Analysis and design of an adaptive queue (AVQ) algorithm for active queue management. In: Proceedings of the ACM SIGCOMM 2001. San Diego, 2001.
- [18] Barakat C. TCP/IP modeling and validation. IEEE Network, 2001,15(3):38~47.
- [19] UCN/LBL/VINT. Network simulator-NS2. 1995. <http://www-mash.cs.berkeley.edu/ns>.
- [20] Ren FY, Wang FB, Ren Y, Shan XM. PID controller for active queue management. Journal of Electronic and Information Technology, 2003,25(1):94~99 (in Chinese with English Abstract).
- [21] Misra V, Gong WB, Towsley D. Fluid-Based analysis of a network of AQM routers supporting TCP flows with an application to RED. In: Proceedings of the ACM/SIGCOMM 2000. Stockholm, 2001.
- [22] Holot C, Misra V, Towsley D, Gong WB. A control theoretic analysis of RED. In: Proceedings of the INFOCOMM 2000. Tel Aviv, 2000.
- [23] Morari G. Internal mode control-1: a unifying review and some new results. Industry and Engineering Chemical Process Design and Device, 1982,21:308~323.
- [24] Rivera DE, Morari M, Skogestad S. Internal mode control-4: PID controller design. Industry and Engineering Chemical Process Design and Device, 1986,25:252~265.

附中文参考文献:

- [20] 任丰原,王福豹,任勇,山秀明.主动队列管理中的PID控制器.电子与信息学报,2003,25(1):94~99.