

文章编号: 1005-8893 (2004) 04-0052-04

几种主动队列管理拥塞控制算法的比较研究

徐燕, 王正洪

(江苏工业学院 计算机科学与工程系, 江苏 常州 213016)

摘要: 主动队列管理是近年来端到端拥塞控制研究的热点, 50多种 AQM 算法已经被提出, 但路由器中采用哪种算法没有统一认识。通过仿真实验, 对 ARED、AVQ、PI 和 REM 4种主动队列管理拥塞控制算法在相同的仿真网络环境中进行了比较研究。实验研究表明, 这4种 AQM 算法都能使队列稳定在目标值, PI 算法使队列最稳定, AVQ 算法维持一个较小的队列长度; 4种 AQM 算法都能经过一定时间适应网络变化的要求, PI 和 AVQ 算法有较好的性能, 短流对 ARED 和 REM 算法有较大影响; 4种算法都没有根本解决对 UDP 流的公平性问题。

关键词: 主动队列管理 (AQM); 拥塞控制; 仿真

中图分类号: TP 393

文献标识码: A

随着 Internet 的发展, 新型网络应用不断拓展, 应用模式不断丰富, 对网络服务质量 (QoS) 提出了更高的要求, 实施拥塞控制是其它 QoS 机制正常工作的必要前提。拥塞控制的研究已经成为网络技术领域研究的重要课题之一^[1]。

Internet 通常依赖于传输控制协议 (TCP) 实现端到端的拥塞控制, TCP 中基于窗口的端到端拥塞控制算法已经成为保证 Internet 稳定性的重要因素。但仅靠源端的控制机制很难提供可靠的 QoS 保证, 增强网络中的路由器等中间节点的功能是实现主机端无法达到技术目标的有效途径。为此, IETF (Internet 工程任务组) 建议采用主动队列管理 (Active Queue Management, AQM) 参与 IP 层拥塞控制。在提出 AQM 的研究时, 由 S. Floyd 和 V. Jacobson 提出的随机早期检测 (Random Early Detection, RED^[2]) 是唯一能实现 AQM 技术目标的算法。近年来, 围绕着 AQM 和 RED 的研究逐渐丰富起来。

1 算法比较与评价

据最近的调查研究表明, 自 1999 年以来, 超

过 50 种新的 AQM 算法已经被提出。在路由器中采用哪种 AQM 算法很难估计, 而且 ISP 对新策略的部署是敏感的。目前, AQM 机制仍然没有被广泛地利用在 Internet 上, 路由器中大多数是基于“弃尾” (Drop-Tail) 队列管理的, RED 算法或 RED 算法的变种 (如: Cisco's WRED) 作为可选配置在路由器上, 但常常不被使用, 因为发现有许多缺点, 甚至有人反对它的发展^[3]。

AQM 的部署步伐之所以慢是由于缺乏对各种算法较为详细的、一致的客观评价标准, 大多数 AQM 评价工作是为了新算法的有效性目的而进行的。通常对 AQM 算法性能的评价主要包括:

(1) 队列的稳定性: AQM 的目的是控制路由器中的队列长度, 因此算法稳定与否直接关系到路由器中队列长度的变化情况, 而队列长度的变化又直接影响到网络的服务质量。一方面, 对于一个特定的 TCP 连接, 由于其传播延迟是固定的, 因此该连接传输时延和时延抖动的大小主要是由路由器中的队列长度所决定的; 另一方面, 路由器中的队列长度直接关系到其输出链路的资源利用率, 只有当队列长度不为零时才能保证网络资源的有效利

收稿日期: 2004-09-29

基金资助: 国家科技部中小企业创新项目资助 (03C26213200933)

作者简介: 徐燕 (1963-), 女, 江苏武进人, 讲师。

用。因此一个好的 AQM 算法应能使队列长度稳定在一个较低的值。

(2) 公平性: AQM 的目标之一是改进 Drop-Tail 队列的公平性。RFC 2309 强调: 路由器的队列机制应保护适应流, 对非适应流进行有效的鉴别和限制。

(3) 适应性: 具有较强的鲁棒性, 即对环境变化不敏感。Internet 的复杂性和异构性决定了网络状态的变化是难以避免的, 因此一个好的 AQM 算法应该对网络状态的变化具有很好的适应能力, 在网络负载、传输时延等因素发生变化时, 仍可实现较好的传输性能。

目前 AQM 主要通过两种参数来估计网络的拥塞状态: (平均) 队列长度和 (平均) 包到达率。

1.1 研究对象

在众多的 AQM 算法中, 选择哪些作为研究对象? 在这里有一个基本原则: ①所选算法具有代表性; ②这些算法能直接地加载到当前的路由器中, 无需修改端节点的 Internet 协议; ③所选算法适合“尽力而为”(best-effort) IP 网络; ④所选算法允许用在高带宽的核心路由器中。

为此选择了 ARED^[4], REM^[5], PI^[6], AVQ^[7] 4 种算法作为研究对象。每种算法都有其特点, 如表 1 所示。

表 1 挑选的 AQM 算法
Table 1 The selected AQM algorithms

AQM 算法	研究手段	拥塞参数	设计主要目标
ARED	启发式	队列长度	性能
REM	乐观主义式	队列长度和包到达率	队列的稳定
PI	控制理论	队列长度	队列的稳定
AVQ	启发式	包到达率	性能

参数设置是在确保评价的一致性和公平性上面面对的主要困难。一般地, 每种算法都有平均 5 个参数, 每个参数都影响算法的性能。上述的每种算法, 一般都提供了参数设置的指导, 但是, 在某种情况下必须推断评价时适用的场合。

1.2 研究方法

评价上述 4 种 AQM 算法采用 NS2 仿真软件, 本研究在 Linux 操作系统上使用 ns-2.27 版本进行仿真实验, 仿真网络的拓扑结构如图 1 所示。两个路由器节点之间为瓶颈链路, AQM 的算法在节点 R1 中实现, 其参数按 4 种算法推荐的参数设置, 其余节点采用 Drop-Tail 机制, 节点 R1 的缓

存大小为 100 个数据包, 数据包长固定为 1 000 bytes, 仿真时间为 200 s。

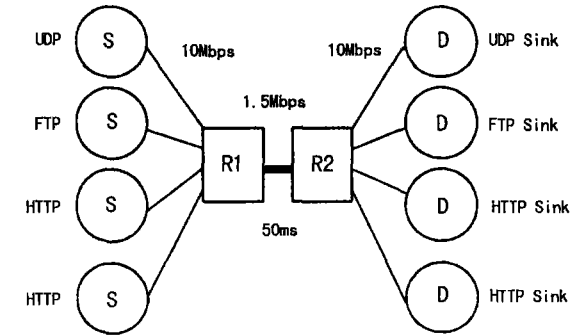


图 1 网络仿真拓扑结构

Fig. 1 Simulation network topology

网络上的数据流为混合流。包括: 持续时间长 TCP 流 (如: 大文件的 FTP 传输), 持续时间短的 TCP 流 (如: 突发流或 Web 流), 非适应流 (如: UDP 流)。

1.3 仿真实验及性能评价

实验 1: 评价队列的稳定性

在这一仿真中, 主要研究 4 种 AQM 的算法是否满足最基本的要求, 即队列的稳定性。为此, 在仿真网络中, 首先, 设计了 50 个 TCP 长流 1s 同时接入, 并一直持续到仿真结束。图 2 (a) (b) (c) (d) 显示了四种算法在 RTT 为 50ms 情况下的仿真情况。可以看出它们基本上能将队列稳定在目标值。其中, PI 算法稳定性最好, AVQ 维持一个较小的队列长度, REM 在稳定队列时, 有一定的延时, ARED 和 REM 都有一定的震荡。

实验 2: 评价动态环境下的鲁棒性

分两组实验, 模拟动态的网络环境。

第一组: 设计 20 个 TCP 长流 1s 接入, 15 个 TCP 长流 75 s 接入, 15 个 TCP 长流 150 s 接入, 一直持续到仿真结束。

第二组: 设计了 20 个 TCP 长流 1 s 接入, 15 个 TCP 短流 75 s 接入 (85 s 结束), 15 个 TCP 短流 150 s 接入 (160 s 结束)。

两组实验的仿真图形基本类似, 图 3 (a) (b) (c) (d) 显示了第二组实验 4 种算法在 RTT 为 50 ms 情况下的仿真结果。可以看出: TCP 短流对每种 AQM 都有影响, ARED、REM 队列有较为强烈的振荡, PI 虽然在网络变化时, 对队列有一定的影响, 但很快恢复控制在目标队列, AVQ 有些变化, 但不明显, 说明 AVQ 的抗短流能力较强。

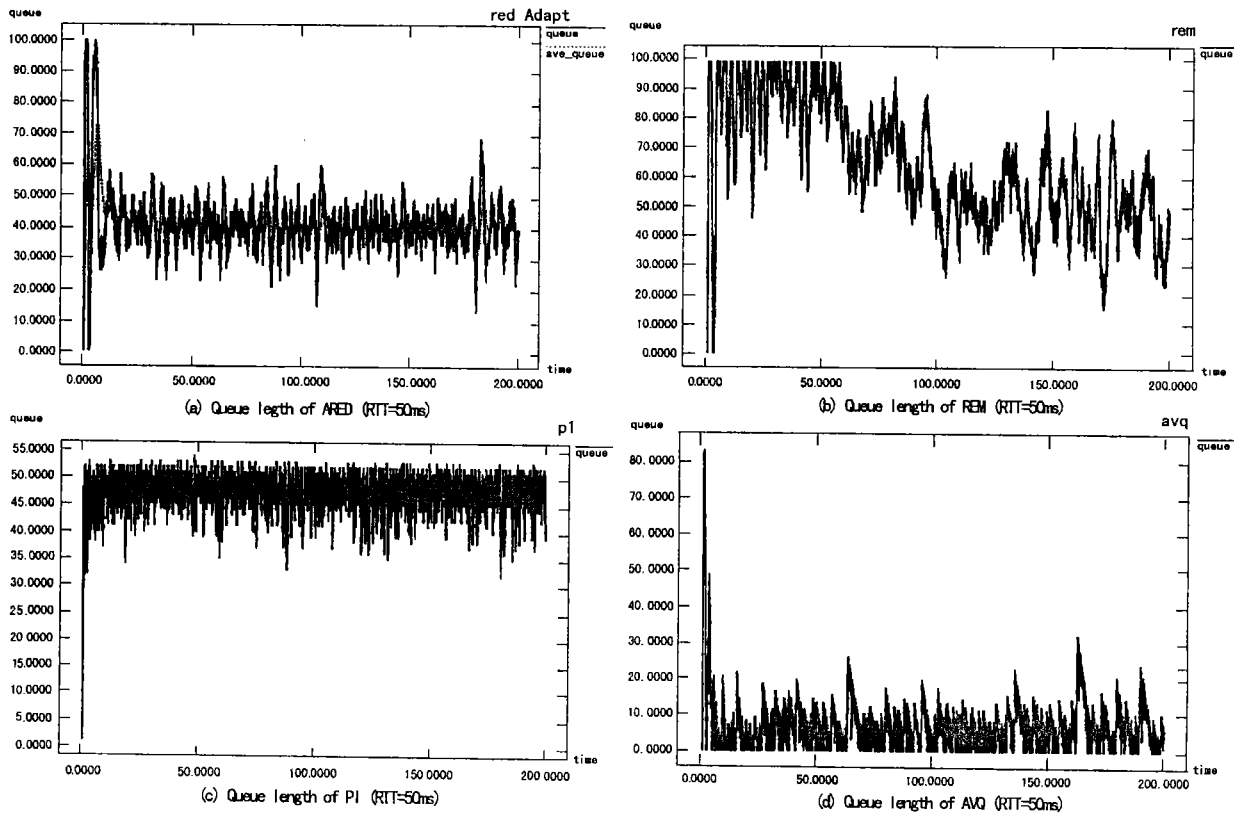


图 2 ARED, REM, PI 和 AVQ 控制下的队列长度变化

Fig. 2 Queue length evolution of ARED, REM, PI and AVQ

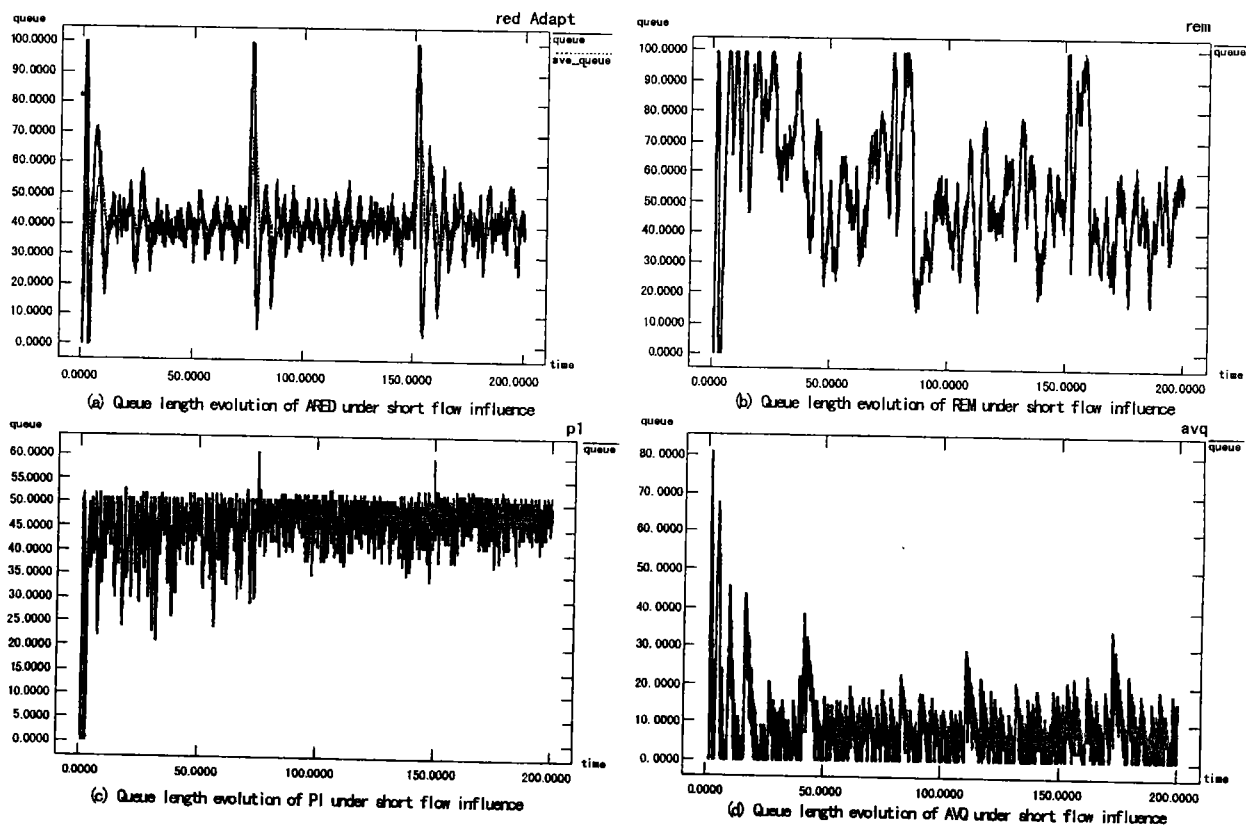


图 3 ARED, REM, PI 和 AVQ 算法在动态环境下的性能

Fig. 3 Queue length evolution of ARED, REM and AVQ in dynamic environment

实验3: 公平性评价

在此实验中, 设计了25个TCP流1s接入, 2个UDP流100s接入, 一直持续到仿真结束。仿真结果显示, 4种算法在UDP流影响下, 它们的性能都比较差。这是因为这4种AQM算法都没有针对UDP流的影响而设计。

实验4: 丢包率统计

本实验测试4种AQM算法在同一仿真条件下的丢包情况。对实验1中, 瓶颈链路RTT分别为100ms, 50ms, 10ms进行测试和对实验2第二组进行测试, 并进行统计。表2显示统计结果。

表2 丢包率统计

Table 2 The percentage statistics of packet dropped

AQM	10 ms	50 ms	100 ms	TCP短流
ARED	2.84%	2.57%	2.15%	1.10%
AVQ	4.08%	3.40%	2.58%	2.26%
PI	3.12%	2.69%	2.26%	1.12%
REM	2.46%	2.17%	1.88%	1.00%

从表中可以看出, 在AQM中当链路延迟增加时, 丢包率下降。

2 结束语

仿真实验表明, 上述4种AQM机制各有其优缺点。四种算法都能使队列稳定在目标值, 达到AQM最基本的目标要求, 但PI算法使队列最为稳定, AVQ算法维持一个较小的队长, 在维持非空队列时也表现得较好, 使得延迟较小、利用率比较高, REM算法需要较长时间才能使队列稳定; 在适应性方面, 4种AQM算法都能经过一定的时间, 适应网络变化的要求, 但PI、AVQ算法有较

好的表现, 而ARED、REM算法有较大的抖动; 在公平性方面, 4种算法在UDP流影响下, 它们的性能都比较差, 都没有根本解决对UDP流的公平性问题。丢包率的统计, ARED和REM有较小的丢包率。在实验中发现参数设置的不同对性能影响较大, 在公平性方面有待于进一步改进。

参考文献:

- [1] 任丰元, 林闯, 刘卫东. IP网络中的拥塞控制[J]. 计算机学报, 2003, 26(9): 1025-1034.
- [2] Floyd S, Jacobson V. Random Early Detection Gateways for Congestion Avoidance[J]. ACM/IEEE Transactions on Networking, 1993, 1(4): 397-413.
- [3] May M, Bolot J, Diot C, et al. Reasons Not to Deploy RED[A]. Proc of 7th International Workshop on Quality of Service (IWQoS'99) [C]. London: IEEE/IFIP, 1999. 260-262.
- [4] Floyd S, Gummadi R, Shenker S. Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management [EB/OL]. <http://www.icir.org/floyd/red.html>, 2001-08-01.
- [5] Athuraliya S, Low S H, Li V H, et al. REM: Active Queue Management[J]. IEEE Network, 2001, 15(3): 48-53.
- [6] Holot C, Misra V, Towsley D, et al. On Designing Improved Controllers for AQM Routers Supporting TCP Flows[A]. Proc of the INFOCOM 2001 [C]. Anchorage Alaska: IEEE Communication Society, 2001. 1726-1734.
- [7] Kunniyur S, Srikant R. Analysis and Design of an Adaptive Virtual Queue Algorithm for Active Queue Management[A]. Proc of the ACM SIGCOMM 2001 [C]. San Diego: ACM Press, 2001. 123-134.

Comparative Study of Several Active Queue Management Algorithms

XU Yan, WANG Zheng-hong

(Department of Computer Science and Technology, Jiangsu Polytechnic University, Changzhou 213016, China)

Abstract: Active Queue Management (AQM) has been a recent focus in the research area of end-to-end congestion control. More than 50 AQM algorithms have been proposed. However, there is no agreement yet which algorithm should be used in network routers. In this paper, four important AQM algorithms, i.e., ARED, AVQ, PI and REM, are compared in the same network environment, by means of network-simulation. The experimental study shows that, while all the studied AQM algorithms can keep queues stable, the queue managed by PI is the most stable one, and AVQ can maintain a shorter queue than ones managed by other algorithms. It is found that, while all the four algorithms can adapt to network conditions, PI and AVQ have better adaptability, and the performance of ARED and REM can be influenced substantially by short-lived TCP flows. Simulation results also show that none of the four algorithms can perfectly solve the unfairness problem for UDP flows.

Key words: Active Queue Management (AQM); congestion control; simulation