

基于扩充的 $G^X/M/1/N$ 排队系统的主动队列管理 算法性能评价模型

汪浩¹ 严伟² 黄明和¹ 郭斌¹

(江西师范大学软件学院 南昌 330022)¹

(北京大学信息科学技术学院光纤通信系统与网络国家重点实验室 北京 100871)²

摘要 为了评价主动队列管理(AQM Active Queue Management)算法的性能,通过在标准的排队系统 $G^X/M/1/N$ 中嵌入主动队列管理(AQM)算法的方法,建立了一个扩充的 $G^X/M/1/N$ 排队系统。利用该扩充的 $G^X/M/1/N$ 排队系统和 Internet 业务流量自相似性的特征,提出了一种评价 AQM 算法在非响应业务流量下的性能的分析方法,并据此评价了 TD, RED, GRED 和 Adaptive RED 等 4 个经典的 AQM 算法。为了验证该分析方法的合理性,利用 NS2 进行了一系列的仿真实验。结果表明,该分析方法所得结果与仿真实验所得结果一致,因此该分析方法能用于评价 AQM 算法的性能。

关键词 计算机网络,主动队列管理(AQM),拥塞控制, $G^X/M/1/N$,输入流稀疏化,自相似业务流量,重尾分布
中图分类号 TP393 文献标识码 A

Extended Queueing System $G^X/M/1/N$ for Evaluating the Performance of AQM Algorithms

WANG Hao¹ YAN Wei² HUANG Ming-he¹ GUO Bing¹

(School of Software, Jiangxi Normal University, Nanchang 330022, China)¹

(State Key Laboratory of Advanced Optical Communication Systems and Networks, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

Abstract In order to evaluate the performance of AQM(Active Queue Management) algorithms, an extended queueing system was developed by embedding an AQM algorithm into the queueing system $G^X/M/1/N$. Using this model and the self-similar traffic of the Internet, a novel approach was proposed to analyze the performance of AQM algorithms with unresponsive traffic. Four classical AQM algorithms TD, RED, GRED and Adaptive RED were assessed by this approach. A series simulation was performed using NS2 to verify the correctness of this approach. The simulation results are consists with those obtained by this approach. This fact shows that the extended queueing system $G^X/M/1/N$ can be used to evaluate the performance of AQM algorithms.

Keywords Computer network, AQM, Congestion control, $G^X/M/1/N$, Thinning of input flows, Self-similar traffic, Heavy-tailed distribution

1 引言

Internet 中的 TCP 主机通过 AIMD(Additive Increase Multiplicative decrease)实现端到端的拥塞控制;但由于 TCP 主机很难了解网络的全局状况,因此希望路由器能预报网络中将要发生的拥塞,并通过主动丢弃 IP 包的方式通知 TCP 主机使其降低发送 IP 包的速率以避免网络发生拥塞。由于 Internet 流量具有突发性,为了吸收突发流量,路由器一般设有用于暂存 IP 包的缓存队列。当缓存队列被 IP 包填满之后,新到达的 IP 包会被路由器丢弃,直到缓存队列腾出空位为止,这种丢弃 IP 包的方式称为尾丢包 TD(Tail Drop)。但

由于 TD 不能及时发现网络中出现拥塞的征兆,因此 IETF 在 RFC2309 中提出了主动队列管理(AQM)算法。AQM 算法的作用是通过监控路由器缓存队列的长度,预测可能发生的拥塞,计算丢包概率(依赖于缓存队列的长度),并依据该丢包概率主动(而不是等缓存队列填满之后)丢弃到达的 IP 包,以配合 TCP 发送端的拥塞控制机制共同对网络实施拥塞控制。RED, GRED 和 Adaptive RED 是三个经典的 AQM 算法^[1-3]。

Internet 业务流由响应流和非响应流组成。长效 TCP 流(如传输大文件的 FTP 应用)称为响应流;短效 TCP 流(如 Web 应用)和 UDP 流(如分布式流媒体应用)称为非响应流。

到稿日期:2008-11-14 返修日期:2009-08-07 本文受国家科技部 973 项目(2009CB320504),国家科技支撑计划(2008BAH37B09),江西省教育厅科技项目(GJJ08153),江西师范大学博士启动基金(基于控制论的主动队列管理算法研究),江西师范大学科研计划项目(一类基于 PI/PID 控制器的 Internet 拥塞控制新算法)资助。

汪浩(1962-),男,博士,教授,主要研究方向为拥塞控制、QoS, E-mail: wanghao@jxnu.edu.cn;严伟(1961-),女,硕士,副教授,主要研究方向为 QoS、流媒体、移动计算;黄明和(1955-),男,教授,主要研究方向为并行计算;郭斌(1970-),男,副教授,主要研究方向为拥塞控制、QoS。

长效 TCP 流的发送端能根据 AQM 的丢包信号控制 IP 包的发送速率;而当 AQM 的丢包信号到达短效 TCP 流的发送端时,短效 TCP 流的发送端通常已无数据可发送,故短效 TCP 流不能响应 AQM 的丢包信号^[4]。UDP 流本身不具备拥塞控制机制,故也不能响应 AQM 的丢包信号。因此,长效 TCP 流称为响应流;短效 TCP 流和 UDP 流称为非响应流。

相关研究 由于长效 TCP 流能响应 AQM 的丢包信号,许多研究者利用闭环模型从理论上分析了长效 TCP 流与 AQM 算法之间的相互作用,并称之为 TCP/AQM 模型^[5-10]。但研究表明,Internet 流量主要由短效 TCP 流(如 Web 流量)构成,其中 65%~80%字节的流量属于短效 TCP 流,55%~75%的 IP 包属于短效 TCP 流^[11,12]。此外,2004 年度 Sigcomm 的获奖者 Simon 认为,随着分布式多媒体应用(如视频会议、视频点播、IP 可视电话、远程教育)的普及,越来越多的 UDP 流将会出现在 Internet 上,并对 Internet 的稳定性产生重要影响^[13]。因此,评价非响应流(短效 TCP 流+UDP 流)对 AQM 算法性能的影响将具有重要意义。

一些研究者通过网络模拟器 NS2 或网络实验分析了非响应流条件下 AQM 算法的性能。Eitan 等人通过模拟手段,分析了短效 TCP 流条件下 RED 算法的公平性^[14]。Christiansen 等人通过实验手段,分析了 RED 算法对 Web 请求响应时间的影响^[15]。Brandauer 等人利用模拟和实验手段分析了 Web 流量条件下 TD,RED 和 GRED 算法的性能^[16]。Iannaccone 等人通过模拟手段,分析了聚合流量条件下,TD,RED 和 GRED 算法的性能^[17]。

由于非响应流(短效 TCP 流+UDP 流)不能响应 AQM 的丢包信号,故另一些研究者试图用开环的排队论模型评价 AQM 算法的性能。Bonald 利用 M/M/1/N 排队系统,评价和比较了 TD 和 RED 算法的性能^[18];Garetto 利用 M^X/M/1/N 排队系统,分析了路由器队列长度(时延)的分布^[19];Younsuk 利用 GI/M/1/N 排队系统,在 IP 包的到达间隔服从 Pareto 分布时,讨论了 TD 算法的丢包率^[20]。

网络测量的统计结果表明,Internet 流量具有自相似性的统计特征,IP 包的到达间隔时间服从重尾 Weibull 分布或 Pareto 分布,并且趋于相互独立和指数分布的随机变量^[21,22]。进一步的研究表明,随着网络流量的增大,到达路由器的 IP 包具有成批到达的特性^[23]。

由于以上原因,我们希望建立一个排队系统,以便考察 Internet 流量自相似性的统计特征对 AQM 算法性能的影响。由于 Bonald 和 Garetto 所用的排队系统都假定到达路由器的 IP 包是 Poisson 流量,故无法利用 Internet 流量自相似性的统计特征。尽管 Younsuk 所用排队系统利用了 Internet 流量自相似性的统计特征;但该排队系统在路由器缓存队列未填满之前将接受到达路由器的每个 IP 包,故不能用于讨论其它 AQM 算法(如 RED 和 GRED)的性能。

本文贡献 由于标准的 GI^X/M/1/N 排队系统在缓存队列未填满前接受到达的每个 IP 包(如图 1 所示),而 AQM 算法需要按一定概率随机丢弃到达的 IP 包,故标准的 GI^X/M/1/N 排队系统不能直接用作 AQM 算法的性能评价模型。本文通过在标准的 GI^X/M/1/N 排队系统中嵌入 AQM 算法(如图 2 所示),建立了一个扩充的 GI^X/M/1/N 排队系统。本文扩展了文献^[18-20]的工作,利用扩充的 GI^X/M/1/N 排队系统

和 Internet 流量自相似性的统计特征,提出了一种评价非响应流对 AQM 算法性能影响的分析方法。在 IP 包的到达间隔时间分别服从重尾 Weibull 分布、Pareto 分布和指数分布的条件下,分别评价比较了 TD,RED,GRED 和 Adaptive RED 等经典 AQM 算法的性能。为了验证该分析方法的合理性和正确性,利用 NS2 进行了一系列的仿真实验;实验结果表明,该分析方法所得结果与仿真实验所得结果完全一致,并且主要结果同其他研究者利用模拟或实验方法所得结果也一致^[16,17],从而验证了该分析方法用于评价 AQM 算法性能的合理性和正确性。



图1 标准的 GI^X/M/1/N 排队系统



图2 扩充的 GI^X/M/1/N 排队系统

本文第 2 节讨论了扩充的 GI^X/M/1/N 排队系统;第 3 节给出了评价 AQM 算法性能的指标;第 4 节利用扩充的 GI^X/M/1/N 排队系统和 Internet 自相似流量评价了 TD,RED,GRED 和 Adaptive RED 等经典算法的性能;第 5 节利用 NS2 仿真实验验证了利用扩充的 GI^X/M/1/N 排队系统评价 AQM 算法的合理性;最后是结论。

2 扩充的 GI^X/M/1/N 排队系统

从本节开始,约定如下记号:

- N:表示排队系统能容纳的 IP 包的个数,其中一个 IP 包正在接受服务,其余的 IP 包($\leq N-1$)在缓存队列等待服务;
- df:表示概率分布函数;
- pdf:表示概率密度函数;
- r. v. :一个随机变量;
- r. v. s:多个随机变量;
- i. i. d:独立同分布。

2.1 扩充 GI^X/M/1/N 排队系统的基本思想

在讨论嵌入 AQM 算法的扩充的 GI^X/M/1/N 排队系统之前,先介绍标准的 GI^X/M/1/N 排队系统(如图 1 所示)。标准的 GI^X/M/1/N 排队系统可表述为^[24]:

1) IP 包成批到达排队系统且每批中 IP 包的个数是随机的。用随机变量 X 表示每批中 IP 包的个数,且 X 具有如下分布: $\Pr(X=j)=g_j$, 其中 j 是正整数;其数学期望记为 $E(X)=\bar{g}$ 。

2) 各批次 IP 包依次在时刻 $\tau_0, \tau_1, \tau_2, \dots, \tau_n, \dots$, 到达排队系统,其中 $\tau_0 < \tau_1 < \tau_2 < \dots < \tau_n < \dots$, 且 $\tau_0 = 0$ 。到达间隔: $\varepsilon_i = \tau_i - \tau_{i-1}$ ($i=1, 2, \dots, n, \dots$) 是独立同分布随机变量,其 df 为 $A(t)$, pdf 为 $a(t)$, 数学期望为 a。

3) 一个服务台,服务台为每个 IP 包提供服务的时间服从参数为 μ 的负指数分布,其 df 记为 $B(t) = 1 - \exp(-\mu t)$ 。服务规则采用先到先服务(FCFS)。

4) 排队系统最多能容纳 N 个 IP 包,其中一个 IP 包正在接受服务,其余的 IP 包($\leq N-1$)在缓存队列等待服务。

5) IP 包到达的间隔时间,服务台的服务时间和每批中 IP 包的个数相互独立。排队系统的输入强度为 $\rho = \frac{\bar{g}}{a\mu}$ 。

6) 由于缓存队列容量有限,当一批 IP 包到达时,如果缓

存队列的剩余空间不够存放该批次的所有 IP 包,则允许该批次的 IP 包进入缓存队列的剩余空间,多余的 IP 包被丢弃。

如果缓存队列未填满,标准的 $GI^X/M/1/N$ 排队系统将接纳到达的每个 IP 包,并将其放入缓存队列中(如图 1 所示),直到将缓存队列填满为止。此时,可以直接利用文献[24]的结果,计算队列长度的概率分布,并据此计算排队系统的性能指标。

但是,在排队系统中嵌入 AQM 算法后(如图 2 所示),当一批 IP 包到达时,即使缓存队列未满载,AQM 算法仍然会依概率随机地丢弃该批 IP 包或随机地允许该批 IP 包进入缓存队列。假设到达排队系统的每批 IP 包的到达间隔的 df 是 $A(t)$,由于 AQM 算法随机地丢弃到达排队系统的各批 IP 包,故真正进入缓存队列的 IP 包变得“稀疏”了,这种现象被称为输入流的稀疏化(Thinning of recurrent flows^[25])。假设到达缓存队列的每批 IP 包的到达间隔的 df 为 $A'(t)$,一般而言 $A'(t) \neq A(t)$,且 $A'(t)$ 依赖于 AQM 算法的丢包概率和 $A(t)$ 。

因此,扩充 $GI^X/M/1/N$ 排队系统的基本思路如下:

(1) 根据 AQM 算法的丢包概率和 $A(t)$,利用输入流稀疏化的方法,找出 $A'(t)$ 和 $A(t)$ 之间的关系,参见式(1)到式(3)。

(2) 利用排队论中的辅助变量法^[24,25] 和 Markov 嵌入链等方法^[24],求解队列长度的概率分布,参见定理 1 和定理 2。

2.2 扩充的 $GI^X/M/1/N$ 排队系统

我们首先给出与 AQM 算法相关的“丢包概率”序列和“收包概率”序列的定义及其作用。

定义 1 称满足条件 $0 = d_0 \leq d_1 \leq \dots \leq d_{N-1} < d_N = 1$ 的序列 $\{d_i\} (i=0,1,\dots,N)$ 为“丢包概率”序列;而称序列 $\{r_i; r_i = 1 - d_i, i=0,1,\dots,N\}$ 为“收包概率”序列。

这两个序列的作用解释如下:假设排队系统中当前有 i 个 IP 包($i=0,1,\dots,N$),当一批新的 IP 包到达时,AQM 算法以概率 d_i 丢弃该批 IP 包;或者说以概率 r_i 允许该批 IP 包进入缓存队列。例如:RED 的丢包概率序列为(此处采用文献[18]的做法,用瞬时队列长度计算丢包概率序列,且同一批次的 IP 包具有相同的丢包概率或收包概率):

$$\begin{aligned} & \{d_i; d_i = 0, \text{ if } 0 \leq i < \min_{th}; \\ & d_i = \frac{i - \min_{th}}{\max_{th} - \min_{th}} \max_p, \text{ if } \min_{th} \leq i < \max_{th}; \\ & d_i = 1, \text{ if } i \geq \max_{th} \} \end{aligned}$$

其中, \min_{th} , \max_{th} , \max_p 为 RED 的输入参数且 $\max_{th} = N$, i 是队列长度。

RED 的收包概率序列为:

$$\begin{aligned} & \{r_i; r_i = 1, \text{ if } i < \min_{th}; \\ & r_i = 1 - \frac{i - \min_{th}}{\max_{th} - \min_{th}} \max_p, \text{ if } \min_{th} \leq i < \max_{th}; \\ & r_i = 0, \text{ if } i \geq \max_{th} \} \end{aligned}$$

基于定义 1,一个嵌入 AQM 算法的扩充的 $GI^X/M/1/N$ 排队系统可表述如下(如图 2 所示),其中 1)到 5)同标准的 $GI^X/M/1/N$ 排队系统,6)修改为:

6)假设缓存队列中当前有 $i (i=0,1,\dots,N)$ 个 IP 包,当一批新的 IP 包到达时,该批 IP 包以概率 r_i 进入排队系统缓存队列,或者说该批 IP 包以概率 d_i 被排队系统丢弃,其中, $r_i + d_i = 1$ 。此时,假设 IP 包成批到达排队系统缓存队列的间隔时间的 df 为 $A_i(t) (i=0,1,\dots,N)$, pdf 为 $a_i(t) (i=0,1,$

$\dots, N)$,均值为 $a_i (i=0,1,\dots,N), t \geq 0$,如图 3 所示。由于缓存队列容量有限,当一批 IP 包进入缓存队列时,如果缓存队列的剩余空间不够存放该批次的所有 IP 包,则允许该批次的 IP 包进入缓存队列的剩余空间,多余的 IP 包被丢弃,如图 2 和图 3 所示。

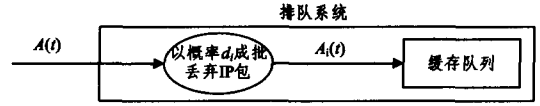


图 3 输入流的稀疏化

按输入流稀疏化的方法, $A_i(t)$ 可以通过 r_i (或 d_i) 和 $A(t)$ 表示为:

1) 当 $i=0,1,2,\dots,N-1$ 时^[25]

$$A_i(t) = \sum_{l=1}^{\infty} r_i (1-r_i)^{l-1} A^l(t) = \sum_{l=1}^{\infty} (1-d_i) d_i^{l-1} A^l(t) \quad (1)$$

其中, $A^l(t)$ 表示 $A(t)$ 的 l 次卷积运算,且 r_i (或 d_i) 和 $A(t)$ 为已知量;式(1)的 Laplace-Stieltjes 变换如下:

$$a_i(s) = \frac{r_i a(s)}{1 - (1-r_i) a(s)} = \frac{(1-d_i) a(s)}{1 - d_i a(s)} \quad (2)$$

其中, $a_i(s)$ 和 $a(s)$ 分别是 $A_i(t)$ 和 $A(t)$ 的 Laplace-Stieltjes 变换。

2) 当 $i=N$ 时, $A_N(t)$ 可用下式估计^[25]:

$$A_N(t) = 1 - e^{-t/a} \quad (3)$$

一旦有了 $A_i(t) (i=0,1,\dots,N)$ 的表达式,可以进一步求解缓存队列长度的概率分布。以下将讨论如何求解该排队系统缓存队列长度的概率分布。先引入两个记号:

$q_k (k=0,1,\dots,N)$: 表示当一个新的 IP 包进入缓存队列前一瞬间,排队系统缓存队列中有 k 个 IP 包的概率;

$p_k (k=0,1,\dots,N)$: 表示任意时刻排队系统缓存队列中有 k 个 IP 包的概率。

我们证明了如下定理:

定理 1 如果已知 $q_i (i=0,1,\dots,N)$, 则计算 p_k 的公式为:

$$p_{k+1} = \frac{1}{\mu} \sum_{i=0}^k \left(\frac{q_i}{a_i} \sum_{j=k-i+1}^{\infty} g_j \right), k=0,1,\dots,N-1 \quad (4)$$

$$p_0 = 1 - \sum_{k=1}^N p_k \quad (5)$$

其中, a_i 为分布函数 $A_i(t) (i=0,1,\dots,N-1)$ 的均值, $1/\mu$ 为排队系统的服务时间(服从指数分布)的均值。

定理 2 $q_k (k=0,1,\dots,N)$ 可以通过求解以下线性方程组获得:

$$q_j = \sum_{i=0}^N q_i h_{i,j}, j=0,1,\dots,N \quad (6)$$

$$\sum_{j=0}^N q_j = 1 \quad (7)$$

计算该线性方程组系数 $h_{i,j}$ 的公式如下所示:

$$h_{i,j} = \begin{cases} \sum_{k=j-i}^{N-i} \beta_{i+k-j}^{(i)} g_k + \beta_{i-j}^{(i)} \sum_{k=N-i+1}^{\infty} g_k, & j > i \geq 0, j \geq 1 \\ \sum_{k=1}^{N-i} \beta_{i+k-j}^{(i)} g_k + \beta_{i-j}^{(i)} \sum_{k=N-i+1}^{\infty} g_k, & 1 \leq j \leq i \end{cases} \quad (8)$$

$$h_{i,0} = 1 - \sum_{j=1}^N h_{i,j} \quad (9)$$

其中,

$$\beta_r^{(i)} = \begin{cases} \int_0^{\infty} \frac{e^{-\mu t} (\mu t)^r}{r!} dA_i(t), & r \geq 0 \\ 0, & r < 0 \end{cases} \quad (10)$$

定理 1 和定理 2 的证明及相关排队系统综述见文献[33,

3 评价 AQM 算法性能的指标体系

如无特别说明,本文后续部分所称“排队系统”就是指经扩充后的“嵌入 AQM 算法的 GI^X/M/1/N 排队系统”。一旦获得排队系统队列长度的概率分布 $q_k (k=0, 1, \dots, N)$ 和 $p_k (k=0, 1, \dots, N)$, 就可以据此计算诸如 IP 包丢失率、链路利用率、队列长度的均值和方差等评价 AQM 算法性能的指标。

3.1 IP 包丢失率的计算公式

当一批新的 IP 包到达时,排队系统根据当前队列长度 $i (i=0, 1, \dots, N)$ 和丢包概率 d_i (或收包概率 r_i) 决定是否丢弃新到达的该批 IP 包。这又分为两种情况。

1) 如果该批 IP 包被丢弃,其丢弃概率为: $q_i d_i$; 故 IP 包丢失率为 $\sum_{i=0}^N q_i d_i$ 。

2) 如果该批 IP 包允许进入缓存队列,由于其中的 IP 包个数可能超出缓存队列的容量,故仍有部分 IP 包要被丢弃,其丢弃概率为: $q_i r_i \sum_{j=N-i+1}^N g_j$; 故 IP 包丢失率为: $\sum_{i=0}^N (q_i r_i \sum_{j=N-i+1}^N g_j)$ 。

如果用 LossRate 表示排队系统的 IP 包丢失率,则:

$$\text{LossRate} = \sum_{i=0}^N q_i d_i + \sum_{i=0}^N (q_i r_i \sum_{j=N-i+1}^N g_j) \quad (11)$$

显然,IP 包丢失率越小越好。

3.2 链路利用率的计算公式

链路利用率表示排队系统缓存队列不空的概率。注意到 p_0 为任意时刻排队系统缓存队列为空(有 0 个 IP 包)的概率;因此,如果用 Utili 表示链路利用率,则:

$$\text{Utili} = 1 - p_0 = \sum_{i=1}^N p_i \quad (12)$$

显然,链路利用率越大越好。

3.3 队列长度的均值与方差的计算公式

通常用队列长度的均值度量延迟,用方差度量延迟抖动。如果用随机变量 ζ 表示任意时刻排队系统缓存队列的长度(不包括正在接受服务的 IP 包),则其均值(平均队列长度)和方差的计算公式为:

$$\bar{\zeta} = \underline{A} E[\zeta] = \sum_{k=1}^N (k-1) p_k \quad (13)$$

$$\text{Var}[\zeta] = \sum_{k=1}^N (k-1-\bar{\zeta})^2 p_k \quad (14)$$

显然,队列长度的均值和方差越小越好。

4 若干经典 AQM 算法的评价结果

我们以三个经典的 AQM 算法 TD, RED, GRED 和 Adaptive RED(简记为 ARED)为例,借助数值分析的手段,分别计算其性能评价指标。计算 AQM 算法性能评价指标的具体步骤如下:

- 1) 对给定的 AQM 算法,确定丢包概率序列 $\{d_i\} (i=0, 1, \dots, N)$ 或收包概率序列 $\{r_i\} (i=0, 1, \dots, N)$ 。
- 2) 根据式(8)到式(10)计算 $h_{i,j}$ (请参见后续的注记)。
- 3) 根据式(6)和式(7)求解 $q_k (k=0, 1, \dots, N)$ 。
- 4) 根据定理 1, 计算 $p_k (k=0, 1, \dots, N)$ 。
- 5) 根据式(11)到式(14)计算各性能指标。

注记:为了用式(8)到式(10)计算 $h_{i,j}$, 首先要找出 $A_i(t)$ 。以下分两种情形讨论之:

1) 如果 $A(t)$ 是参数为 λ 的负指数分布函数,根据式(2)可以求得 $A_i(t)$ 是参数为 $r_i \lambda$ 的负指数分布函数,即 $A_i(t) = 1 - e^{-r_i \lambda t}$ 。

2) 如果 $A(t)$ 是 Weibull, Pareto 等重尾分布函数,可以根据式(2),利用“数值 Laplace 变换”和“数值 Laplace 逆变换”^[30-32], 得到 $A_i(t)$ 的密度函数的数值结果;或者根据重尾分布函数的性质和式(1)得到 $A_i(t)$ 的渐近表达式^[27,28]。

按表 1 的参数设置,分别对路由器缓存队列为 $N=30$ 和 $N=50$ 两种情况,计算了 TD, RED, GRED 和 ARED 算法在 IP 包的到达间隔时间分别服从 Pareto, Weibull 和指数分布时的性能指标,其中随机变量 X 在不同输入强度下的分布参见文献[23]。受篇幅限制,仅给出 $N=50$ 时的计算结果。先给出 TD, RED, GRED 和 ARED 算法在 IP 包的到达间隔时间服从 Weibull 分布时的性能曲线,然后给出 RED 算法在到达间隔时间分别服从 Pareto, Weibull 和指数分布时的性能曲线。图例中的 TD_B, RED_B, GRED_B 和 ARED_B 分别表示 TD, RED, GRED 和 ARED 算法。

表 1 TD, RED, GRED 和 ARED 算法的参数设置

	N=30			N=50		
	max _p	min _{th}	max _{th}	max _p	min _{th}	max _{th}
TD	NA	NA	NA	NA	NA	NA
RED	0.1	10	30	0.1	15	50
GRED	0.1	10	20	0.1	15	35
ARED	0.5	10	30	0.5	10	50

4.1 IP 包到达间隔时间服从 Weibull 分布时, TD, RED, GRED 和 ARED 的性能比较

4.1.1 TD, RED, GRED 和 ARED 算法的 IP 包丢失率

通过观察图 4 中 TD, RED, GRED 和 ARED 算法的 IP 包丢失率的折线图,可以发现:

- 1) 随着输入强度的增加, TD, RED, GRED 和 ARED 算法的 IP 包丢失率逐渐增加。
- 2) 随着输入强度的增加, TD, RED, GRED 和 ARED 算法的 IP 包丢失率没有明显区别。

也就是说与 TD 相比, RED, GRED 和 ARED 不能降低 IP 包丢失率,此结论也被其他作者的模拟和实验结果所证实^[16,17]。

4.1.2 TD, RED, GRED 和 ARED 算法的链路利用率

通过观察图 5 中 TD, RED, GRED 和 ARED 算法链路利用率的折线图,可以发现:

- 1) 随着输入强度的增加, TD, RED, GRED 和 ARED 算法的链路利用率逐渐增加。
- 2) 随着输入强度的增加, TD, RED, GRED 和 ARED 算法的链路利用率没有明显区别。

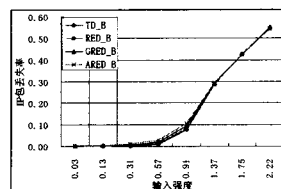


图 4 IP 包丢失率

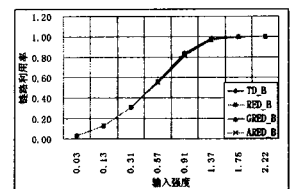


图 5 链路利用率

也就是说与 TD 相比, RED, GRED 和 ARED 不能提高链路利用率,此结论也被其他作者的模拟和实验结果所证实^[16,17]。

4.1.3 TD, RED, GRED 和 ARED 算法的队列长度的均值与方差

平均队列长度是延迟的一种度量。通过观察图 6 中 TD, RED, GRED 和 ARED 算法的平均队列长度的折线图, 可以发现:

1) 随着输入强度的增加, TD, RED, GRED 和 ARED 算法的平均队列长度逐渐增加。

2) 在不同的输入强度下:

TD 平均队列长度 > RED 平均队列长度 > GRED 平均队列长度 > ARED 平均队列长度, 也就是说, AQM 算法在一定程度上降低了端到端的延迟, 该结论与文献[16, 17]的结论一致。

类似地, 队列长度的方差是延迟抖动的一种度量, 通过观察图 7 中 TD, RED, GRED 和 ARED 算法的队列长度方差的折线图, 可以发现:

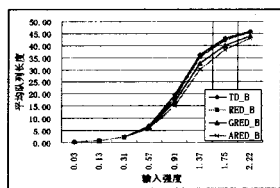


图 6 平均队列长度(均值)

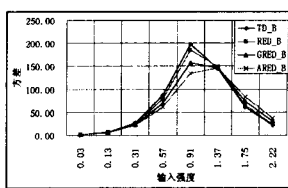


图 7 队列长度的方差

1) 随着输入强度的增加, 当输入强度小于 1 时, TD, RED, GRED 和 ARED 算法队列长度的方差逐渐增加; 当输入强度大于 1 时, TD, RED, GRED 和 ARED 算法队列长度的方差逐渐减小。这表明当输入强度逐渐增加但小于 1 时, 由于平均队列长度较小, 瞬时队列长度围绕平均队列长度振荡的幅度逐渐增大; 当输入强度逐渐增加且大于 1 时, 由于平均队列长度较大(接近 $N=50$), 瞬时队列长度围绕平均队列长度振荡的幅度逐渐减小。

2) TD, RED, GRED 和 ARED 算法的队列长度的方差有一些差异。当输入强度小于 1 时, TD 的方差 > RED 的方差 > GRED 的方差 > ARED 的方差; 当输入强度大于 1 时, TD 的方差 < RED 的方差 < GRED 的方差 < ARED 的方差。

4.2 IP 包到达间隔时间分别服从 Pareto, Weibull 和指数分布时, RED 的性能比较

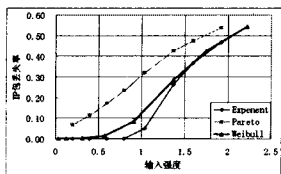


图 8 不同分布下 RED 的 IP 包丢失率

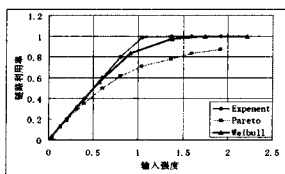


图 9 不同分布下 RED 的链路利用率

因篇幅所限, 仅给出当 IP 包到达间隔时间分别服从 Pareto, Weibull 和指数分布时, RED 算法的 IP 包丢失率和链路利用率, 如图 8 和图 9 所示。TD, GRED 和 ARED 算法也有类似结果。从图上可知, Pareto 具有最大(最小)的 IP 包丢失率(链路利用率); Exponent 具有最小(最大)的 IP 包丢失率(链路利用率); Weibull 夹在它们中间。

5 仿真实验结果

为了验证扩充的 $G^X/M/1/N$ 排队系统用于评价 AQM

算法在非响应业务流量下的性能的合理性, 用 NS2 做了一系列的仿真实验, 并和第 4 节的分析方法所得结果进行对比, 这是许多研究 AQM 模型的文献的习惯做法, 见文献[5-10, 18]。

仿真实验采用与文献[5-10, 14, 18]类似的哑铃型网络拓扑(如图 10 所示)。注意到, 做仿真实验时, 瓶颈链路 N-D 上“IP 包丢失率和链路利用率”对应图 2 中扩充的 $G^X/M/1/N$ 排队系统的“IP 包丢失率和链路利用率”。

仿真实验采用的哑铃型网络拓扑描述如下: 若干个源节点通过瓶颈链路 N-D 与若干个目标节点相连, 源节点中有若干个发送 TCP 业务流量的 TCP 节点和发送 UDP 业务流量的 UDP 节点。瓶颈链路 N-D 的带宽为 1.8Mbps, 传播时延为 1ms, 队列长度为 50packets; 其它链路的带宽均为 100 Mbps, 传播时延为 1m。为了仿真实验的 Web 流量(采用与文献[14]相同的方法), 每个 TCP 节点依次发起 100 个短效 TCP 会话, 短效 TCP 会话之间的发起间隔时间服从负指数分布, 平均间隔时间为 0.9 秒; 并且每个 TCP 会话传输的文件大小服从 Pareto 分布, 平均大小为 10kb~20kb(这也是 Web 页面的平均大小)。TCP 客户端使用 Reno 或 NewReno。UDP 节点从模拟开始至结束一直发送 CBR 流量。通过控制 TCP 节点和 UDP 节点的数量, 可以生成具有不同输入强度的业务流量, 其中 TCP 节点产生的业务流量占 80%~90%, UDP 节点产生的业务流量占 10%~20%。为了便于与第 4 节分析方法所得结果相比较, 仿真实验中 TD, RED, GRED 和 ARED 的参数设置与第 4 节相同。以下是各次仿真实验的其它一些参数设置。

表 2 各次仿真实验的其它一些参数设置

TCP 节点数	UDP 节点数	文件平均大小	TCP 客户端
3	1	10kb	Reno
20	4		
50	6		
80	9	15kb	Newreno
100	12		
120	13	20kb	

针对不同输入强度的业务流量(共 6 种不同强度的业务流量)、传输文件的平均大小(共 3 种不同的平均大小)、TCP 客户端(共 2 种不同的客户端)和不同的 AQM 算法(TD, RED, GRED 和 ARED), 总共做了 $6 * 3 * 2 * 4 = 144$ 次模拟实验。通过分析仿真结果, 获得了 IP 包丢失率、链路利用率、队列长度的均值与方差等 AQM 算法的性能指标。以下是部分仿真实验的结果。

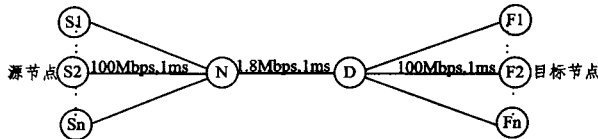


图 10 网络拓扑结构

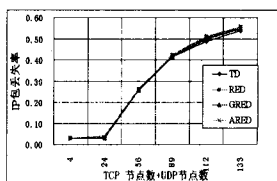


图 11 IP 包丢失率(10kb, Newreno)

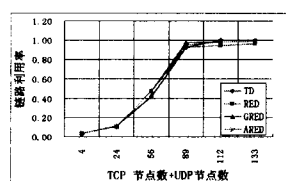


图 12 链路利用率(10kb, Newreno)

图 11 和图 12 分别为文件平均大小=10kb, TCP 客户端为 Newreno 时, 仿真实验所得到的相应算法的 IP 包丢失率和链路利用率。图 13 和图 14 分别为文件平均大小=20kb, TCP 客户端为 Reno 时, 仿真实验所得到的相应算法的 IP 包丢失率和链路利用率。

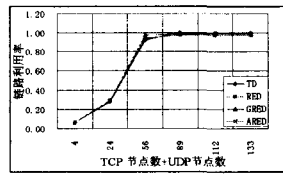
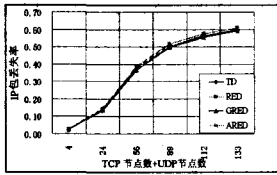


图 13 IP 包丢失率(20kb, Reno) 图 14 链路利用率(20kb, Reno)

通过观察图 11 和图 13 中 TD, RED, GRED 和 ARED 算法的 IP 包丢失率的折线图可以发现, 随着输入强度(TCP 节点数+UDP 节点数)的增加, TD, RED, GRED 和 ARED 算法的 IP 包丢失率逐渐增加; 但 TD, RED, GRED 和 ARED 算法的丢包率之间没有明显区别。该仿真实验的结果与图 4 利用分析方法所得结论一致。

通过观察图 12 和图 14 中 TD, RED, GRED 和 ARED 算法的链路利用率的折线图可以发现, 随着输入强度的增加, TD, RED, GRED 和 ARED 算法的链路利用率逐渐增加, 并且 TD, RED, GRED 和 ARED 算法的链路利用率之间没有明显区别。该仿真实验的结果与图 5 利用分析方法所得结果一致。

结束语 本文利用排队论中输入流稀疏化的方法, 在标准的 $G^X/M/1/N$ 排队系统中嵌入 AQM 算法, 建立了一个扩充的 $G^X/M/1/N$ 排队系统; 给出了该排队系统的 IP 包丢失率、链路利用率、队列长度的均值/方差等性能评价指标。利用该扩充的 $G^X/M/1/N$ 排队系统和 Internet 业务流量自相似性的特征, 提出了一种评价 AQM 算法在非响应业务流量下的性能的分析方法, 并据此评价了 TD, RED, GRED 和 Adaptive RED 等 4 个经典的 AQM 算法。为了验证该分析方法的合理性, 利用 NS2 进行了一系列的仿真实验, 仿真实验所得结果与分析方法所得结果一致。并且, 该分析方法所得结果同其他研究者利用模拟或实验方法所得结果也一致。以上事实说明了该分析方法能用于评价 AQM 算法的性能。

参考文献

[1] Floyd S, Jacobson V. Random Early Detection Gateways for Congestion Avoidance [J]. IEEE/ACM Transactions on Networking, 1993, 11(4): 397-413

[2] Floyd S. Recommendations on using the gentle variant of RED [EB/OL]. <http://www.aciri.org/floyd/red/gentle.html>, Mar. 2000

[3] Floyd S. Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management [EB/OL]. <http://www.icir.org/floyd/papers.html>, Aug. 2001

[4] Youngmi J, Vinay R, Anja F. On the impact of Variability on the buffer dynamics in IP networks [C]//Proc. 37th Annual Allerton Conf. Commun., Control Comp. Sept. 1999

[5] Misra V, Gong W, Towsley D. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED [J]. Proc. SIGCOMM'2000. ACM, 2000, 30(4):

151-160

[6] Mario B, Alfio L, Giovanni A. A fluid-based model of time-limited TCP flows [J]. Computer Networks, 2004, 44(3): 275-288

[7] Tinnakornsriruphap P, Makowski A M. Limit behavior of ECN/RED gateways under a large number of TCP flows [C]//Proc. INFOCOM'2003. IEEE, 2003, 873-883

[8] Tinnakornsriruphap P, Richard J L. Characterization of Queue Fluctuations in Probabilistic AQM Mechanisms [C]//Proc. SIGMETRIC'2004. ACM, 2004, 283-294

[9] Jae C, Mark C. Analysis of Active Queue Management [C]//Proc. of the 2nd IEEE Int. Symposium on Network Computing and Applications. IEEE, 2003

[10] Alazemi H M, Mokhtar A, Azizoglu M. Stochastic approach for modeling random early detection gateways in TCP/IP networks [C]//Proc. ICC 2001

[11] Brownlee N. Understanding Internet Traffic Streams: Dragonflies and Tortoises [J]. IEEE Communications Magazine, 2002, 40(10): 110-117

[12] Thompson K, Miller G J, Wilder R. Wide-Area Internet Traffic Patterns and Characteristics [J]. IEEE Network, 1997, 11(6): 10-23

[13] Simon R. Keynote Speech by SIGCOMM Award Winner Simon Lam [EB/OL]. http://www.acm.org/sigs/sigcomm/sigcomm2004/conf_program.html, 2004

[14] Eitan A, Tania J. Simulation analysis of RED with short lived TCP connections [J]. Computer Network, 2004, 44(5): 631-641

[15] Christiansen M, Jeffay K, Ott D, et al. Tuning RED for Web Traffic [J]. IEEE/ACM Tran. on Networking, 2001, 9(3): 249-269

[16] Brandauer C, Iannaccone G, Diot C, et al. Comparison of Tail Drop and Active Queue Management Performance for bulk-data and Web-like Internet Traffic [C]//Proc. 6th IEEE Symposium on Computers and Communications. IEEE, 2001, 3-5

[17] Iannaccone G, May M, Diot C. Aggregate Traffic Performance with Active Queue Management and Drop from Tail [J]. Proc. SIGCOMM'2001. ACM, 2001, 31(3): 4-13

[18] Bonald T. Analytic evaluation of RED performance [C]//Proc. INFOCOM'2000. IEEE, 2000, 1415-1424

[19] Garetto M, Towsley D. Modeling, Simulation and Measurements of Queuing Delay under Long-tail Internet Traffic [C]//SigMetrics'2003

[20] Younsuk K, Kiseon K. Loss Probability Behavior of Pareto/M/1/K Queue [J]. IEEE Communication Letters, 2003, 7(1): 39-41

[21] Cao J, Cleveland W S, Dong L, et al. On the nonstationarity of Internet traffic [J]. Proc. SIGMET-RICS'2001, ACM, 2001, 29(1): 102-112

[22] Tudjarov A, Temkov D, Janevski T, et al. Empirical modeling of Internet traffic at middle-level burstiness [C]//Proc. 12th IEEE Mediterranean. IEEE, 2004, 535-538

[23] Cao J, Cleveland W S, Lin D, et al. The Effect of Statistical Multiplexing on the Long-Range Dependence of Internet Packet Traffic [R]. Bell Labs, 2001

[24] Vijaya P L, Gupta U C. Analysis of finite-buffer multi-server queues with group arrivals; $G^X/M/c/N$ [J]. Queueing System, 2000, 36(1-3): 125-140

[25] Vladimir V K. Mathematical Methods in Queuing Theory [M]. Kluwer Academic Publishers, 1994

[26] Finch P D. Balking in the queueing system GI/M/1 [J]. Acta Math. Acad. Sci. Hungar., 1959, 10: 241-247

[27] Sigman K. Appendix : A Primer on Heavy - tailed Distributions [J]. Queueing Systems, 1999, 33(1-3): 261-275

[28] Charles M, Claudia K. Subexponential Distributions [M]//J. A. Robert, et al., eds. A practical guide to heavy tails: statistical techniques and applications. Boston: Birkhauser, 1998

[29] Takacs L. On a Combined Waiting Time and Loss Problem Concerning Telephone Traffic [J]. Ann. Univ. Sci. Budapest. Eotvos. Sect. Math., 1958, 1: 73-82

[30] Shortle J F, Brill P H, Fischer M J. An Algorithm to Compute

the Waiting Time Distribution for the M/G/1 Queue [J]. INFORMS Journal on Computing, 2004, 16(2): 152-161

[31] Piessens R, Huysmans R. Algorithm 619 Automatic Numerical Inversion of the Laplace Transform [J]. ACM Trans. on Mathematical Software, 1984, 10(3): 348-353

[32] Abate J. Numerical Inversion of Laplace Transforms of Probability Distribution [J]. ORSA Journal on Computing, 1995, 7(1): 36-43

[33] 汪浩. 一个具有随机丢弃分组机制且分组成批到达的 GI^X/M/1/N 排队系统[J]. 数学的实践与认识, 2005, 35(9): 113-120

[34] 汪浩, 李晓明, 严伟. 一类具有随机离去策略的排队系统及其在 Internet 拥塞控制中的应用[J]. 计算机科学, 2007, 34(12): 37-40

(上接第 152 页)

当电动机处于正转过程中, 停止按钮按下, 库所 p_0 通过变迁 t_{12} 到达库所 p_{10} (库所 p_0 消耗两个托肯, 库所 p_{10} 获得两个托肯), 电动机正转停止; 电动机反转过程中按下停止按钮与上同理可得。当停止按钮重新按下时, 库所 p_{10} 通过变迁 t_{13} 到达库所 p_0 (库所 p_{10} 消耗两个托肯, 库所 p_0 获得两个托肯)。

5 模型分析

使用 Petri 网作为建模工具时, 需要考虑的基本性质有^[10]:

1. 有界性, 刻画状态空间的有限性;
2. 活性, 与所有可达标识下的潜在发生权有关。无死锁是一个弱化的条件, 它指网系统模型的全局的无限动作能力 (即动作发生), 但它的某个部分可能根本不工作;
3. 可逆性, 刻画从任何可达的标识到初始标识的可恢复性;
4. 互斥性, 刻画子标识不可能同时成立或变迁不可能并发执行的特征。

下面以第 3 节中的电动机正反转控制模型为例说明上述性质。

1. 前述网模型中具有标识为 $p_0 - p_{10}$ 的 11 个表示系统状态的库所节点, 因此状态空间大小有限, 有界性条件满足;

2. 网模型中从初始态 (库所 p_0 具有两个托肯, 库所 p_1 具有一个托肯), 根据外界开关量的输入可以实现电动机正转、反转和停止运动三种不同的运动方式, 并且在任一状态只要存在开关量的输入网模型便可以无限运行下去, 因此具有活性特征;

3. 在初始态下, 库所 p_0 具有两个托肯, 库所 p_1 具有一个托肯, 在电动机正转的情况下, 操作员若希望切换到反转状态, 首先就要断开正转启动按钮, 这时网模型中库所 p_8 通过变迁 t_6 和 t_7 恢复库所 p_0 和 p_1 中消耗的托肯返回到初始状态, 同理电动机从反转切换到正转时也要经过类似的过程, 因此系统的可逆性在程序逻辑允许的情况下得到满足;

4. 当正转启动按钮被操作员按下时, 网模型中库所 p_1 通过变迁 t_0 到达库所 p_2 , 库所 p_1 中的托肯消耗为零, 在这种条件下库所 p_1 不可能通过变迁 t_1 到达库所 p_5 , 因此避免了电动机正反转开关同时闭合的可能, 系统互斥性根据控制程序

逻辑得到满足。

结束语 本文利用 Petri 网技术建立了 PLC 控制程序的执行模型。在 PLC 控制程序的运行过程中动态执行 Petri 网模型, 当网模型执行完毕时控制程序结束运行, 这样如果编程人员或用户需要了解控制程序的具体执行过程, 可以将网模型的执行过程记录在指定的文件中, 从而满足程序使用者的特定需求。另外在控制程序执行前通过检测 Petri 网模型具有的活性、互斥性等性质, 可以发现部分逻辑错误以减少调试时间。虽然该执行模型具有上述优点, 但会增加控制程序执行所需的时间, 如果控制程序属于严格实时的任务, 必须最大限度减少执行 Petri 网模型所需的时间, 因此如何优化 Petri 网模型以适应实时的要求是今后进一步的研究方向。

参考文献

- [1] Yang Gong - yuan . Theory and Application of Programmable Logic Controller [M]. Beijing: Publishing House of Electronic and Industrial, 2004
- [2] Yan Ying - fu . Foundation of Programmable Logic Controller [M]. Beijing: Publishing House of People's Postal, 2005
- [3] Xie Ke - ming . Theory and Program Design of Programmable Logic Controller [M]. Beijing: Publishing House of Electronic and Industrial, 2003
- [4] Petri C A . Communication with automata [D] . University of Bonn, Bonn, 1962
- [5] Reisig W . Petri nets, an introduction [A] // Brauer W , Rozenberg G , Salomaa A , eds. EATCS, Monograph on Theoretical Computer Science [C]. Berlin: Springer Verlag, 1985
- [6] Zurawski R, Zhou Mengchu. Petri nets and industrial applications; a tutorial [J]. IEEE Transaction on Industrial Electronics, 1994, 41(6): 567-583
- [7] Yuan Chong-yi. Theory and Application of Petri Net [M]. Beijing: Publishing House of Electronic and Industrial, 2005
- [8] Zaitsev D A. Invariant of timed petri nets [J]. Cybernetics and Systems Analysis, 2004, 40(2): 226-237
- [9] van Aalst W, van Hee K. Workflow management models, methods and systems [M]. MIT Press, 2002
- [10] Girault C, Valk R. Petri Net of System Engineering [M]. Beijing: Publishing House of Electronic and Industrial, 2005