

文章编号:1006-3080(2009)03-0457-05

基于加强型价格的随机指数标记算法

汪 浩¹, 牛玉刚²

(1. 上海交通大学自动化系, 上海 200240; 2. 华东理工大学信息科学与工程学院, 上海 200237)

摘要: 随机指数标记算法(REM)是一种有效的主动队列管理算法,但存在队列收敛性差、对网络环境变化响应慢等问题。通过理论分析表明REM算法类似于PI控制策略,感知网络拥塞的能力不足。为克服上述缺点,提出了基于加强型价格的随机指数标记算法(EPREM)。EPREM在原价格中增加数据包到达速率的变化率,得到加强型的价格,同时引入带宽缓存比例因子指导新增参数 λ 的设定。在NS2中的仿真实验结果表明:相对于REM算法,EPREM提高了收敛速率,增强了算法对网络环境变化的适应性。

关键词: 网络拥塞控制; 主动队列管理; 随机指数标记; 价格; 带宽缓存比例因子

中图分类号: TP393

文献标志码:A

Random Exponential Marking Algorithm Based on Enhanced Price

WANG Hao¹, NIU Yu-gang²

(1. Department of Automation, Shanghai Jiaotong University, Shanghai 200240, China;

2. School of Information Science and Engineering, East China University of
Science and Technology, Shanghai 200237, China)

Abstract: Random exponential marking (REM) is an effective algorithm for active queue management (AQM), but suffers from some drawbacks such as slow convergence speed, and slow sluggish response to dynamic network traffic. Theoretical analysis shows that REM has similar property to PI scheme which could not detect congestion effectively. To overcome these shortcomings, an enhanced REM algorithm, EPREM, is proposed, in which an enhanced price is added and bandwidth-buffer proportion factor is introduced to tune the parameter λ . The simulation experiments based on NS2 show that EPREM can achieve faster convergence speed and better adaptive to variable network conditions.

Key words: network congestion control; active queue management; random exponential marking; price; bandwidth-buffer proportion factor

近年来,计算机网络爆炸式的增长导致网络拥塞问题越来越严重,因此,实现网络拥塞控制的机制和算法受到广泛关注。1998年,互联网工程任务组(IETF)建议在路由器中采用主动队列管理(AQM)机制作为端到端拥塞控制的一种增强手段^[1]。

AQM的思想是对拥塞进行早期检测,向终端发出拥塞指示,终端通过相应的机制在路由器队列溢出和丢包之前降低数据包的发送速率,从而在减小排队时延的同时保证较高的吞吐量。

具有较大影响的AQM算法有RED^[2]、ARED^[3]、

收稿日期:2008-06-24

基金项目:国家自然科学基金(60674015);上海市重点学科建设项目(B504);上海市教委科研创新重点项目(09zz60)

作者简介:汪 浩(1986-),男,安徽黄山人,硕士生,研究方向为网络拥塞控制。E-mail:hwang_seice@sjtu.edu.cn

通讯联系人:牛玉刚,E-mail:acniuyg@ecust.edu.cn

PI^[4]、PID^[5]、REM^[6]和AVQ^[7]等。其中随机指数标记算法(REM)利用Kelly提出的网络流量优化理论中“价格”概念来探测和控制网络的拥塞状态^[6,8]。标准REM算法具有静态网络性能与拥塞度量解耦的优点,但是仿真结果表明:其队列收敛速率慢,对网络环境突变响应不及时,不能保证服务质量。

为了克服REM算法的缺陷,本文利用控制论的方法分析了REM出现上述问题的原因,提出了一种改进的REM算法,即基于加强型价格的随机指数标记算法(EPREM)。EPREM引入数据包到达速率的变化率得到加强型的价格,同时引入带宽缓存比例因子的概念,指导新增参数的设定。

1 REM 算法

1.1 REM 算法

REM算法用影子价格作为拥塞度量。价格变化由两方面因素决定:一方面是瞬时队列长度与目标队列长度的差值,另一方面是报文到达速率与链路带宽的差值,两者的加权和作为价格。通过计算价格大小,并将其嵌入到可以被源端检测到的端到端数据包标记概率中,从而调整源端的发送速率。

价格的迭代公式为:

$$p_l[(k+1)T] = [p_l(kT) + \gamma(\alpha(b_l(kT) - b_l^*) + x_l(kT) - c_l(kT))]^+$$

其中 $\alpha > 0$ 和 $\gamma > 0$ 是加权参数; $[z]^+ = \max\{z, 0\}$; $b_l(kT)$ 和 b_l^* 分别是 kT 时刻队列 l 的长度和目标队列长度; $x_l(kT)$ 是 kT 时刻进入队列 l 的输入量,即报文到达速率; $c_l(kT)$ 是链路带宽。 $x_l(kT) - c_l(kT)$ 度量速率的匹配情况, $b_l(kT) - b_l^*$ 度量队列长度的匹配情况。

单个链路的标记概率是该链路价格数值的指数形式,端到端标记概率是该条路径上的所有拥塞链路价格之和的指数形式。假设一个包经过链路 l , $l=1, 2, \dots, L$,各链路在 kT 时刻的价格为 $p_l(kT)$,则在时刻 kT ,队列 l 处的标记概率为 $m_l(kT) = 1 - \Phi^{-p_l(kT)}$,其中 $\Phi > 1$ 为常数。

数据包的端到端标记概率为:

$$1 - \prod_{l=1}^L [1 - m_l(kT)] = 1 - \Phi^{-\sum_L p_l(kT)}$$

其中 $\sum_L p_l(kT)$ 是指数据包通过链路的价格之和。

1.2 REM 算法分析

对REM算法价格公式从理论上进行简单的分析,以期得出REM算法的一些特点。令 $cm(kT) = \gamma[\alpha(b_l(kT) - b_l^*) + x_l(kT) - c_l(kT)]$,记为瞬时拥

塞度量。标准REM算法价格迭代公式为 $p_l[(k+1)T] = p_l(kT) + cm(kT)$,且初始时刻时 $p_l(0) = 0$,所以 kT 时刻的价格为:

$$\begin{aligned} p_l(kT) &= p_l[(k-1)T] + cm[(k-1)T] = \\ &\sum_{i=0}^{k-1} cm(iT) = \sum_{i=0}^{k-1} \gamma[\alpha(b_l(iT) - b_l^*) + x_l(iT) - \\ &c_l(iT)] = \sum_{i=0}^{k-1} [\beta e(iT) + \gamma \Delta e(iT)] = \\ &\beta \sum_{i=0}^{k-1} e(iT) + \gamma e(kT) \end{aligned}$$

其中: $\beta = \gamma\alpha$, $e(iT)$ 为队列长度误差, $\Delta e(iT) = e[iT] - e(iT) = x_l(iT) - c_l(iT)$ 为队列长度误差变化量。

标准REM算法数据包丢弃或标记概率为:

$$\begin{aligned} P(kT) &= 1 - \Phi^{-p_l(kT)} \approx 1 - [1 - (\ln \Phi) p_l(kT)] = \\ &(\ln \Phi) p_l(kT) = \ln \Phi (\beta \sum_{i=0}^{k-1} e(iT) + \gamma e(kT)) = \\ &\beta \sum_{i=0}^{k-1} e(iT) + \gamma e(kT) \end{aligned}$$

其中: $1 - \Phi^{-p_l(kT)} \approx 1 - [1 - (\ln \Phi) p_l(kT)]$ 是对 $\Phi^{-p_l(kT)}$ 进行迈克劳林展开,并由于 $\Phi = 1.001$ 以及实际情况下的 $(\ln \Phi) p_l(kT) \ll 1$,故而舍弃级数的高次项而得到的。另外 $\beta' = (\ln \Phi) \beta$, $\gamma' = (\ln \Phi) \gamma$ 。

通过分析不难发现,REM 可以近似视为 PI 控制, β' 和 γ' 分别是积分和比例环节的系数。

正是由于REM算法类似于PI控制机制,从而导致其对环境变化的响应能力较差,队列收敛速率较慢。例如,在仿真实验中,0 s 时刻处激活 N 个TCP连接发送数据,这时队列长度首先经历小于设定值的过程,队列长度误差 $e < 0$ 。REM 算法在其比例环节的作用减小价格,表明没有拥塞发生或者拥塞缓解,进而减小标记概率,队列长度必将迅速上升;只有当队列长度超过设定值,且越来越大时,REM 才感知到拥塞加剧,而这时相对于有限的缓存而言,有大量的数据流进入,很可能已经导致队列溢出;或者在相反的情况下,队列排空,链路利用率下降。由此可见,AQM 算法的响应能力对保证网络服务质量十分重要。

2 EPREM 算法

2.1 带宽与缓存比例因子

路由器作为一种存储转发设备,其内部缓存能够提高链路使用率,减少路由器丢包数。一方面,缓存过小可能导致其应对突发数据流的能力不足,造成大量强制性丢包;另一方面,缓存的增加将会增大时延,降低网络性能。路由器缓存大小的需求分析已经成为当前 Internet 的重要研究领域^[9]。如今

在主动队列管理算法的研究中,常用的瓶颈链路带宽配置为0.5~15 Mb,路由器缓存大小为50~800 pkts,每个数据包的大小为500~1 000 bytes。

路由器中队列长度的波动是数据包到达速率与链路带宽存在差值而引起的。现在一般路由器缓存为几十兆,而相对于百兆的链路带宽,缓存显得相对较小。特别是在互联网的骨干网中,缓存更加无法与带宽匹配。在目前大多数的仿真实验中,路由缓存配置的大小也不及带宽的配置。

基于上述分析,本文引进一个衡量网络环境配置的因子——带宽与缓存比例因子 μ :

$$\mu = 1024^2 C / BMT$$

其中: C 为链路带宽,Mb/s; B 为路由器缓存,pkts; M 为数据包大小,bytes; $T=8$ bit/byte。

比例因子 μ 衡量的是带宽与缓存数值的差距大小,也就是在没有拥塞控制策略以及队列管理算法作用时,1 s内数据流量相对于带宽的波动将对队列长度相对于路由缓存产生多大程度的影响。

μ 对后面提出的EPREM算法参数 λ 的设定有一定的指导意义。因为在 μ 很大的网络环境中, λ 设置不能过大,否则将造成队列振荡;如果 μ 较小, λ 设置不能过小,否则微分环节的作用体现得不明显。即满足关系: $\lambda \propto 1/\mu$ 。

2.2 EPREM 算法描述

一旦网络发生拥塞时,数据包到达速率一定大于瓶颈链路带宽,如果负载越重,起初超过带宽的流量就越大。为了讨论方便,取报文到达速率为带宽大小 C ,通过一组常见的网络仿真环境来计算带宽缓存比例因子 μ 。选择 $C=5$ Mb, $B=150$ pkts,数据包大小 $M=500$ bytes,计算得 $\mu \approx 8.74$,即相对于带宽1倍的数据流速波动,在1 s内就可能造成队列相对于缓存8.74倍波动的潜在影响。如果队列调整不及时,就可能出现队列长度溢出,造成强制性丢包,或者队列排空,导致链路利用率下降等问题。反过来说,如此大的流量如果能够得到控制,必将对AQM算法的实施效果带来益处,比如提高响应能力,获得更稳定的队列。

REM算法对拥塞的到来没有预知能力,在如今 μ 普遍较大的网络环境中,无法保证良好的网络性能。为了提高主动队列管理算法的队列稳定性以及对动态数据流的响应速度,必须加强对数据包到达速率的管理。本文通过在价格中引入数据包到达速率的变化率,将REM算法的PI控制结构改进为PID控制策略,得到基于加强型价格的随机指数标记算法。在标准REM算法瞬时拥塞度量的基础

上,增加数据包到达速率变化率,得到EPREM的瞬时拥塞度量为:

$$cm'(kT) = \gamma\{a[b_l(kT) - b_l^*]\} + x_l(kT) - c_l(kT) + \lambda[x((k+1)T) - x(kT)]$$

其中: $x((k+1)T) - x(kT)$ 为 kT 时刻数据包到达速率变化率。参数 γ 和 a 与标准REM中一致, λ 为新增参数。

EPREM中加强型价格为:

$$q_l((k+1)T) = q_l(kT) + cm'(kT) = q_l(kT) + \beta[b_l(kT) - b_l^*] + \gamma[x_l(kT) - c_l(kT)] + kd[x((k+1)T) - x(kT)]$$

其中: $\beta = \gamma a$, $kd = \gamma \lambda$ 。

EPREM算法的标记或丢弃概率为:

$$Q(kT) = 1 - \Phi^{-q_l(kT)} \approx (\ln \Phi) (\beta \sum_{i=0}^{k-1} e(iT) + \gamma e(kT) + kd \Delta e(kT))$$

通过对当前网络配置的分析,本文指出流量变化率控制的必要性;同时通过控制理论的方法对标准REM算法的分析,指出其缺陷,故而引入流量变化率得到加强型的价格。如果仅仅从标准REM算法提出的角度去分析,即基于优化理论提出最优化流控模型,把发送率看作原始变量,把拥塞度量看成是对偶变量,现有的TCP/AQM算法看作是使资源利用率最大的Lagrangian方法,很难找到算法缺陷及相应的改进策略。

2.3 EPREM 算法实现

EPREM算法核心部分的伪代码如下:

```

e = q - q_ref * ;
de = e - e_old ;
pr = pr_old + γ[αe + (in - c) + λ(de - de_old)] ;
e_old = e ;
de_old = de ;
if (pr < 0.0) { pr = 0.0 ; }
p = 1.0 - pow(phi, - pr) ;
pr_old = pr ;

```

把编写好的EPREM.h和EPREM.cc文件放在NS2目录下,修改NS中的ns-default.tcl和makefile文件,然后重新编译,在系统没有提示错误后,表明模块添加成功。

3 算法性能评价

3.1 仿真模型

在NS2中构建如图1所示的网络拓扑结构。

它是由 N 个发送端 $S_1 \sim S_n$ 和 N 个接收端 $D_1 \sim D_n$ 以及路由器 R_1, R_2 组成的哑铃型结构。

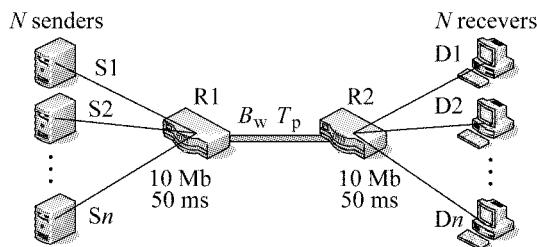


图 1 仿真网络拓扑结构

Fig. 1 Simulation network topology

每条链路上标记有带宽、传输延时。其中瓶颈链路带宽 B_w , 传输延时为 T_p ; 路由器缓存为 200 pkts, 目标队列长度为 100 pkts; 用户数目为 N 个; 其他全部链路带宽为 10 Mb, 传输延时 T_p 为 50 ms。路由器中的 AQM 算法分别采用 REM 和 EPREM 进行比较。REM 和 EPREM 设置相同的参数, $\gamma = 0.001$, $\alpha = 0.1$, $\Phi = 1.001$; 另外参数 λ 的设定在不同实验中有所不同。在仿真过程中, 源端 S_i 向对应的接收端 D_i 发送大量的 FTP 数据, 分组大小为 500 bytes。

3.2 实验结果分析与讨论

3.2.1 静态网络环境下的队列变化对比 首先设定网络环境为: 瓶颈链路带宽 1 Mb, TCP 连接数 100, T_p 为 5 ms。比较 REM 和 EPREM 对应的队列变化。EPREM 算法中 λ 为 0.3。

如图 2 所示, 相对于 REM 算法, EPREM 算法的队列收敛速率更快。REM 队列在 80 s 后才达到目标队列值, 而 EPREM 队列在 40 s 前达到稳态, 明显优于 REM 队列收敛性; 同时 EPREM 队列稳定性也比 REM 队列好。图 3 中, EPREM 的丢包率调整明显快于 REM, 可以看出 EPREM 对拥塞发生有更快的响应。

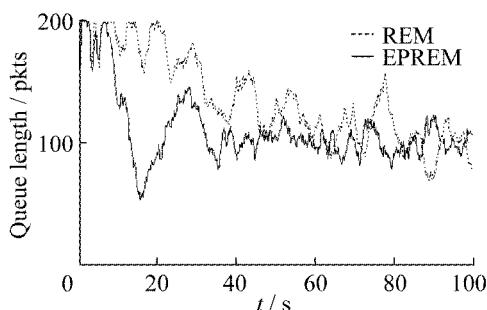


图 2 REM 和 EPREM 队列长度变化情况

Fig. 2 Queue evolution of REM and EPREM

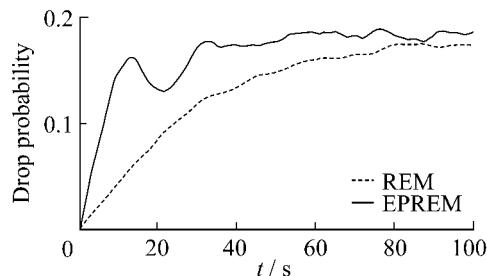


图 3 REM 和 EPREM 丢包率变化情况

Fig. 3 Probability evolution of REM and EPREM

3.2.2 TCP 连接数突变时的队列变化对比 取瓶颈链路带宽为 1 Mb, 建立 120 个 TCP 连接, 仿真 300 s。在 0 s 时激活 60 个连接直至仿真结束, 在 100 s 时激活另外 60 个连接, 在 200 s 时关闭。EPREM 算法中 λ 为 0.3。

TCP 连接数变化是网络环境变化的最常见的原因之一。算法对用户数变化是否能快速响应, 队列长度是否能快速到达设定值, 并保持稳定是算法优劣的重要指标。在图 4 中, EPREM 队列长度在 TCP 连接数突变时, 较 REM 更快地将队列长度重新调整回到目标数值, 有效地减少了因队列溢出引起的强制丢包, 以及由于路由队列清空而引起的链路利用率下降。图 5 为 TCP 连接数突变时丢包率对比, 在 0, 100, 200 s 3 个时刻, EPREM 的丢包率调整明显比 REM 的调整快。

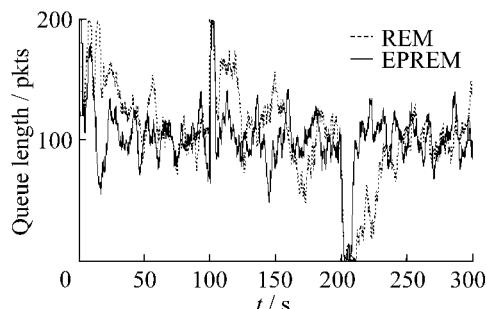


图 4 REM 和 EPREM 队列长度变化情况

Fig. 4 Queue evolution of REM and EPREM

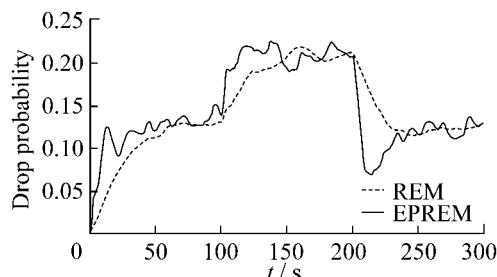


图 5 REM 和 EPREM 丢包率变化情况

Fig. 5 Probability evolution of REM and EPREM

3.2.3 混合流下的队列变化对比 取瓶颈链路带宽为 1 Mb, 建立 100 个 TCP 连接以及一条 CBR 数据流。UDP 流在 100 s 时启动, 流经瓶颈链路, 导致 FTP 数据流带宽的突减; 200 s 时关闭 CBR 数据流。CBR 数据流的发送间隔为 0.009 s, 数据大小为 500 bytes。EPREM 算法中 λ 为 0.3。

CBR 流的引入必然占用瓶颈链路的带宽, 而带宽作为重要的网络资源, 其变化必然导致队列的波动。如图 6 所示, 100 s 处 REM 和 EPREM 队列激增, 200 s 处队列突减, 但是 EPREM 拥有更快的调整速度。在图 7 中也可以看出 EPREM 丢包率调整快于 REM。

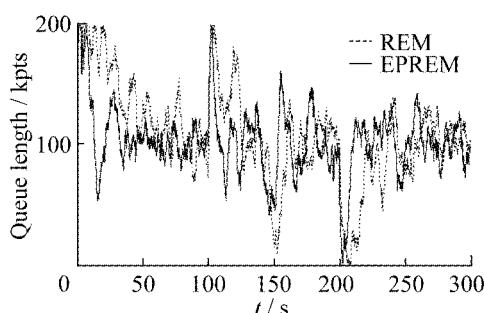


图 6 REM 和 EPREM 队列长度变化情况
Fig. 6 Queue evolution of REM and EPREM

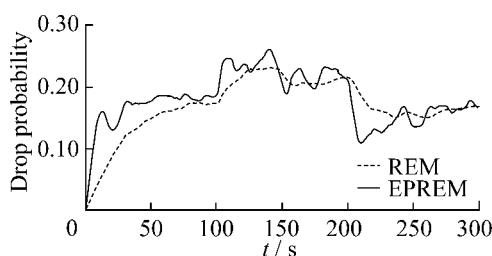


图 7 REM 和 EPREM 丢包率变化情况
Fig. 7 Probability evolution of REM and EPREM

3.2.4 带宽缓存比例因子 μ 对参数 λ 设定的指导 在仿真实验的基础上, 增加瓶颈链路带宽为 2 Mb, 采用 REM、EPREM 算法进行仿真, 其中 REM 算法参数不变, EPREM 算法中 λ 分别为 0.3 和 0.15, 结果见图 8。

如图 8 所示, 当增大瓶颈链路带宽至 2 Mb 时, 仍取 EPREM 中参数 λ 为 0.3, EPREM($\lambda=0.3$) 队列收敛虽然较 REM 快, 但是其振荡却明显大于 REM 队列。重新计算此时的带宽与缓存比例因子 μ 约为 2.6, 这时根据 $\lambda \propto 1/\mu$, 计算得 $\lambda=0.15$ 。从仿真结果中不难看出此时的 EPREM($\lambda=0.15$) 队列抖动明显减小, 提高了网络性能, 有力地说明了带宽与缓存比例因子 μ 对参数 λ 设定的指导意义。

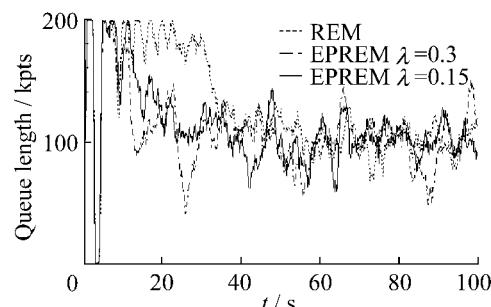


图 8 REM, EPREM($\lambda=0.3, 0.15$) 队列长度变化情况
Fig. 8 Queue evolution of REM and EPREM ($\lambda=0.3, 0.15$)

4 结 论

针对 REM 算法的缺陷, 分析了导致这些问题的原因。提出了一种基于加强型价格的随机指数标记算法(EPREM), 通过大量的仿真实验(分别在静态网络环境、动态网络环境、TCP、UDP 混合流的网络环境中进行仿真)将 REM 与 EPREM 进行对比, 结果表明: EPREM 较 REM 具有更快的队列收敛速率, 同时有效减小由于队列溢出而引起的强制性丢包, 减少由于路由队列排空造成的吞吐量下降的情况, 提高了算法在复杂环境中的适应性。

参 考 文 献:

- [1] Braden B, Clark D. Recommendations on queue management and congestion avoidance in the internet [C]//RFC 2309 Internet performance Recommendation. [s. l.]: IEEE, 1994: 1-17.
- [2] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance [J]. IEEE/ACM Transactions on Networking, 1993, 1(4): 397-413.
- [3] 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/>, 2001.
- [4] Hollot C V, Misra V, Towsley D, et al. On designing improved controllers for AQM routers supporting TCP flows [C]//Proc IEEE INFOCOM 2001. USA: IEEE Communication Society, 2001: 1726-1734.
- [5] 任丰原, 王福豹, 任勇, 等. 主动队列管理中的 PID 控制器[J]. 电子与信息学报, 2003, 25(1): 94-99.
- [6] Athuraliya S, Low S H, Li V H, et al. REM: Active queue management [J]. IEEE Network, 2001, 15(3): 48-53.
- [7] Srisankar S Kunnumur, Srikant R. An adaptive virtual queue (AVQ) algorithm for active queue management [J]. IEEE/ACM Transactions on Networking, 2004, 12(2): 286-299.
- [8] Sanjeeva Athuraliya, Steven Low. Price computation in random early marking [C]//IEEE International Conference on Communications 2000. USA: IEEE Communication Society, 2000: 485-489.
- [9] 李玉峰, 邱菡, 兰巨龙, 等. 路由器缓存需求 [J]. 软件学报, 2008, 19(3): 733-743.