

## 一种支持区分服务的模糊公平分组丢弃算法

陈 远 李乐民

(电子科技大学宽带光纤传输与通信网技术重点实验室 成都 610054)

**摘 要** 随机早期检测(RED)算法是目前 Internet 中广泛使用的一种缓存管理算法,对 RED 算法进行改进以提高公平性以及网络动态变化的适应性一直是主动队列管理的研究重点之一。该文引入新的变量——分享指数,取代单个业务流缓存占用量来表征不同业务流对网络资源的占用情况,以克服缓存占用量表征公平性的缺陷。同时,在无线网络中对分享指数的定义进行推广,使其同时兼顾公平性和系统性能(信道条件)。通过对分享指数和队列长度共同制定规则,提出了一种基于模糊逻辑控制的缓存管理算法。算法在根据缓存队列长度计算丢弃概率时,根据不同流对网络资源的占用情况(即分享指数)进一步决定增加或减小相应的丢弃概率,从而可利用不同流之间的差异以加速缓解拥塞,具有更好的自适应性和公平性。仿真表明,采用该算法后,系统性能得到改善。

**关键词** 缓存管理, 分组丢弃, 随机早期检测, 模糊逻辑控制, 公平性

中图分类号: TN915.07

文献标识码: A

文章编号: 1009-5896(2006)06-1129-06

## A Fuzzy Fair Packet Dropping Algorithm Supporting Differentiated Services

Chen Yuan Li Le-min

(Key Lab of Broadband Optical Fiber Transmission and Communication Networks, UESTC, Chengdu 610054, China)

**Abstract** Random Early Detection (RED) is a kind of buffer management algorithms which is widely used in the current Internet. To improve the adaptability and fairness of RED is one of the main tasks of the active queue management. In this paper, the buffer occupancy of individual flow is replaced by a new introduced variable – *Sharing Index* to indicate the network resource share among different flows, so as to overcome the limitation of buffer occupancy. The definition of *Sharing Index* in wireless networks is also extended to make it consider both fairness and system performance (channel condition). Through defining rules of both *Sharing Index* and queue length, a fuzzy logic based buffer management algorithm named FF-RED is proposed. When calculating packet dropping probability in FF-RED algorithm, the probability will further increase or decrease based on the sharing of network resource (by means of *Sharing Index*), so it can use the difference between flows to make congestion relief more quickly, and it has better adaptability and fairness. Simulation results show that the system performance is improved through using the proposed algorithm.

**Key words** Buffer management, Packet dropping, Random Early Detection (RED), Fuzzy Logic Control (FLC), Fairness

### 1 引言

在Internet中,随机早期检测(Random Early Detection, RED<sup>[1]</sup>)算法是一种广泛使用的缓存管理算法。它基于平均队列长度预测可能的网络拥塞,并采用随机丢弃(或标记)的策略,在拥塞尚未出现时提示发端降低发送速率,以达到避免拥塞的目的。多年的实践证明,RED算法具有较好的性能。

然而RED算法具有下述两大明显的缺陷:(1)基于best-effort服务,没有考虑不同等级服务之间、不同用户流之间的差别,无法提供有效的公平性保障。RIO<sup>[2]</sup> (RED with

In/Out bit)算法虽然提供了对多业务的支持,但仍然没有解决相同优先级的业务流之间的公平性。(2)对控制参数较敏感,难以优化参数设定,算法的性能对控制参数和网络流量负载的变化非常敏感。在用户流增大的情况下,RED算法的性能会下降。如何对算法进行改进是主动队列管理中重要的研究内容。

由于RED在特定时刻对所有流使用同一丢弃概率,这对低速的业务流和分组较短的业务流显然不公平,丢弃高速业务流的分组比丢弃低速业务流分组更能够加速拥塞缓解,同时也有利于公平的分配网络资源。考虑到对每一个业务流进行速率估计需要较大的计算量,因此文献中大多采用简化的方法,一种是通过丢弃分组进行计数以确定大量占用网络

2004-10-08 收到, 2005-03-10 改回  
国家自然科学基金委员会与香港研究资助局联合科研基金(6021800  
2)资助课题

资源的业务流,另一种方法是利用不同业务流占用的缓存空间量来近似表征其速率的相对大小。常见的考虑公平性的缓存管理算法中,BRED<sup>[3]</sup>(Balanced RED)基于每个业务流的缓存占用量对速率较高的流施以较大的丢弃概率以改善公平性;FRED<sup>[4]</sup>(Flow RED)算法虽然对不同业务流分组采用同样的丢弃概率,但对缓存占用量大的业务流分组施以较大的强制丢弃概率。

采用上述两种考虑优先丢弃高速业务流分组的算法最大的好处是公平性的改善。公平性的好处是显而易见的:能尽量满足每个业务流的QoS要求;能对业务流进行隔离,防止非响应流抢占带宽;能一定程度上对突发业务进行平滑,抑制其对缓存管理算法的负面影响。然而用缓存占用量近似表征速率的有效性值得怀疑。当业务流较少时,缓存占用量能近似反应业务流速率的相对大小,但随着业务流数目的上升,这种有效性会逐渐降低,当业务流数目与缓存空间大小(以分组为单位)可比拟时,上述两种公平性算法将失去作用。而采用对丢弃分组计数的方法在响应流(如TCP)与非响应流(如UDP)共存时,会出现非响应流抢占带宽的情况。因此公平性算法同RED算法一样需要解决鲁棒性和自适应性。

我们把文献中对提高RED自适应性的研究分成3类:第1类算法认为线性的丢弃概率和固定的门限值而造成RED性能不稳定的根本原因。Adaptive-RED<sup>[5]</sup>算法根据网络拥塞程度相应地增大或减小丢弃概率以提高RED对网络流量负载变化的自适应能力。Stabilized-RED<sup>[6]</sup>,BLUE<sup>[7]</sup>等算法使用类似的思想。第2类算法认为采用控制理论对缓存管理进行分析,使之能够在高吞吐量和低时延之间做出平衡,维持队列长度在期望值之内。典型的算法如比例积分控制器(PI)<sup>[8]</sup>,随机指数标记(REM)<sup>[9]</sup>等。第3类认为由于网络的动态特性及负载的变化,缓存管理算法的参数难于确定,也很难适应网络的变化,对于这类具有动态变化特性,难于建立精确分析模型的系统采用模糊逻辑进行控制能取得较好的结果,文献[10-13]对采用模糊逻辑控制进行缓存管理进行了研究。

现有的模糊逻辑算法大多以队列长度和队列长度变化作为输入对缓存进行模糊控制,从控制的角度来说,利用差值作为反馈是最直观有效的方法,也能够较快地达到稳定。虽然模糊逻辑能适应网络的动态变化,对多个业务流有很好的适应性,但它与RED算法一样,并没有考虑不同业务流之间的差别。

基于以上考虑,本文提出了一种基于模糊逻辑控制的缓存管理算法——FF-RED(Fuzzy Fair - Random Early Detection)算法。我们引入一个新的变量——分享指数,取代单个业务流缓存占用量来表征不同业务流对网络资源的占用情况,以克服缓存占用量表征公平性的缺陷。同时,考虑到无线网络的一些特点,对无线网络中分享指数的定义进行推广,使其同时兼顾公平性和系统性能(信道条件)。FF-RED算法对分享指数和缓存队列长度共同制定模糊规则,可利用不同流之间

的差异以加速缓解拥塞,具有更好的自适应性和公平性。

## 2 系统模型及算法描述

图1显示了模糊逻辑控制缓存管理算法的系统模型,其中包括3个主要模块:Diff-Serv优先级分类、RED处理和模糊逻辑控制器(FLC)。通过识别IP包头的DS字段,优先级分类将属于不同类别连接的分组排入相应的优先级队列。FLC对每个优先级队列提取各自的参数,在模糊化后以语言变量的形式输入FLC,FLC根据相互独立的模糊规则推理出每个队列的丢弃概率模糊值,再通过反模糊化得到精确的丢包概率。

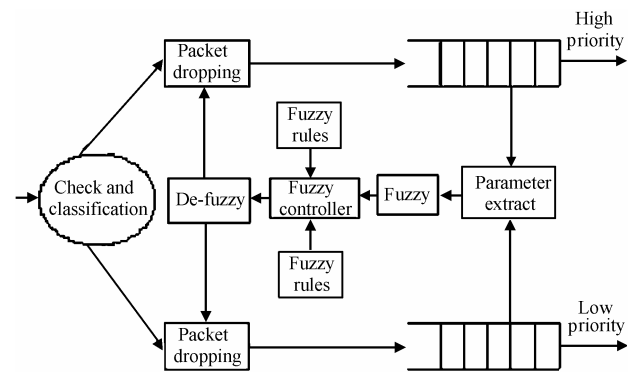


图1 系统模型

Fig.1 System model

对于模糊逻辑算法来说,最重要的是如何定义模糊规则及选择参数进行控制。缓存管理算法的目的是根据网络的拥塞情况对速率进行调节,因此模糊规则的制定也是围绕拥塞判定来进行的。大多数的缓存管理算法对拥塞判定都是基于队列长度(如RED),少数则基于速率进行拥塞判定(如AVQ),还有一些则二者兼顾(如PI)。现有文献中的模糊逻辑算法大多基于第1类,即利用队列长度和长度变化制订模糊规则,算法的最终目的是将队列长度控制在参考值附近。然而这类算法并未考虑不同流之间的差异,丢弃(或标记)高速业务流和低速业务流的分组对拥塞缓解的程度是完全不一样的。我们认为,超过其比例占用网络资源的业务流可近似看做高速业务流,优先丢弃高速业务流,不仅能达到公平性,也能快速缓解拥塞。因此我们提出的模糊公平缓存管理算法希望能以队列长度和单个业务流速率的相对大小共同制订模糊规则。

### 2.1 分享指数 SI(Sharing Index)

考虑到对聚合流中的每一个业务流进行速率监控实现相对复杂,因此我们沿用文献中常用的利用缓冲空间占用量来表征不同业务流速率相对大小的思想,为提高算法的适应性和鲁棒性,我们对此方法进行改进,引入一个新的参数——分享指数(SI)来表征不同业务流速率的相对大小:

$$SI_i = \frac{avg_i - QLen_{ref} \cdot weight_i}{avg_i + QLen_{ref} \cdot weight_i} \quad (1)$$

$$\text{avg}_i = (1 - \omega) \cdot \text{avg}_i + \omega \cdot \text{qlen}_i \quad (2)$$

其中 $\text{qlen}_i$ 为业务流 $i$ 占用的缓存空间大小,  $\text{avg}_i$ 为业务流 $i$ 的平均缓存占用,  $\text{QLen}_{\text{ref}}$ 为希望维持的队列长度参考值,  $\text{weight}_i$ 为不同业务流的归一化权值, 当所有业务流具有相同权值时,  $\text{weight}_i = 1 / \text{FlowNumber}$ ,  $0 \leq i \leq \text{FlowNumber}$ , 此时式(1)可改写为

$$\text{SI}_i = \frac{\text{avg}_i - \text{QLen}_{\text{ref}} / \text{FlowNumber}}{\text{avg}_i + \text{QLen}_{\text{ref}} / \text{FlowNumber}} \quad (3)$$

由于我们没有采用按流排队的模型, 而是所有相同优先级的业务流共享一条FIFO缓存队列, 因此 $\text{qlen}_i$ 只是在该业务流分组到达或离开时进行简单的计数操作, 并没有子队列长度的含义, 我们称之为单个业务流的缓存占用量。

在计算单个业务流平均缓存占用时, 我们采用与RED算法中计算平均队长类似的指数窗口滑动平均(EWMA)方法。虽然采用同样的方法, 但RED中计算的平均队长只具有队列长度的含义, 并不能表征速率的大小; 而在我们的算法中,  $\text{avg}_i$ 更接近采用滑动窗口对一定时间内业务流到达分组进行计数, 它的大小反应了不同业务流之间速率的相对大小。

在RED计算平均队长时, 权值 $\omega$ 一般固定的取0.002, 这一权值在计算聚合流中单一业务流到达分组的滑动平均时并不一定合适。在业务流数目较小时, 突发业务对缓存管理的影响更大一些, 我们希望对较大的权值 $\omega$ , 此时计算的 $\text{avg}_i$ 要能快速地反映当前时刻的缓存占用情况; 而在业务流数目较大时, 较小的 $\omega$ 能更精确的计算平均队长, 从而提高以平均缓存占用量表征单一业务流速率大小的有效性。因此在计算平均缓存占用量时我们根据业务流数目对权值 $\omega$ 进行动态调整, 取 $\omega = 1 / (2k \cdot \text{FlowNumber})$ , 则计算平均缓存占用的公式写为

$$\text{avg}_i = \left(1 - \frac{1}{2k \cdot \text{FlowNumber}}\right) \cdot \text{avg}_i + \frac{1}{2k \cdot \text{FlowNumber}} \cdot \text{qlen}_i \quad (4)$$

其中 $k$ 为大于0的整数。为了对当前时刻系统资源的公平分配进行快速地反应, 我们取 $k=1$ 。 $k$ 越大则计算越准确, 从长的时间尺度上看也越公平, 但同时系统状态变化的反应也相对滞后。

显然, 从式(1)可得 $\text{SI}_i \in (-1, 1)$ 。当 $\text{SI}_i$ 接近-1时, 表示业务流 $i$ 在最近一段时间里到达分组很少, 几乎未占用任何网络资源; 随着 $\text{SI}_i$ 的增大, 该业务流到达分组逐渐增多, 业务流速率也逐渐增大, 但只要 $\text{SI}_i < 0$ , 业务流 $i$ 对网络资源的占用仍小于其分配的比例; 当 $\text{SI}_i = 0$ 时, 表示该业务流按照规定的比例分享网络资源; 而 $\text{SI}_i > 0$ 时, 表示业务流 $i$ 的到达速率已大于按比例分配给它的网络带宽,  $\text{SI}_i$ 越大, 该业务流对网络资源的过度占用越严重。在进行拥塞控制时, 降低具有较大 $\text{SI}_i$ 的业务流的发送速率, 不仅是公平的, 也是最快有效的。

当业务流数目较多时, SI表征公平性比缓存队列占用量

更为准确(考虑公平性的缓存管理算法和公平排队(FQ)算法一般都对并发业务流数目有一定限制)。然而其缺点也比较明显, 由于算法需要计算每个业务流的平均平均缓存占用量, 运算复杂度高于RED算法。我们认为, 随着网络设备软硬件计算能力的提高, 算法复杂度是可以接受的。

## 2.2 无线网络中分享指数SI的推广

在无线网络中同样需要合理的策略对缓存进行有效管理, 以尽可能避免和缓解拥塞, 从而保证服务质量。然而无线链路与有线链路相比具有较大的特殊性, 对资源的公平分配显得更为复杂。继续沿用式(1)的分享指数定义并不一定是合理的。例如, 假设基站中同时有两个具有相同优先级的用户, 一个用户的信道条件很差, 链路速率降到16 kbps, 而另一个用户具有优良的信道条件, 链路速率达到384 kbps, 如果此时考虑公平性, 为两个用户维护相同的缓存队列长度, 给信道条件较差的用户平等的传输机会, 显然系统性能会很差, 对信道条件较好的用户不公平。因为对信道条件较好的用户来说, 它所获得的系统分配给它的传输时间只有前者的1/24。为了使算法具有更广泛的适应性, 我们对式(1)进行推广, 使分享指数的定义从公平的分配缓存空间资源转换到较公平的分配时间资源, 使之能应用于无线网络之中。在设计算法时, 我们希望能允许当前信道条件较好的用户占用超过其比例的缓存空间, 而信道条件较差的用户暂时让出部分空间。即尽可能在用户信道条件较好时允许分组进入缓存。

因此, 我们将式(1)改为

$$\text{SI}_i = \frac{\text{avg}_i - \text{QLen}_{\text{ref}} \cdot \text{weight}_i \cdot \alpha}{\text{avg}_i + \text{QLen}_{\text{ref}} \cdot \text{weight}_i \cdot \alpha} \quad (5)$$

引入一个新的变量 $\alpha$ 作为衡量信道条件的参数。在有线网络中, 由于信道条件不会发生变化, 因此 $\alpha$ 固定为1, 此时式(5)蜕变为式(1); 当信道条件较好时 $\alpha > 1$ ,  $\text{QLen}_{\text{ref}} \cdot \text{weight}_i \cdot \alpha$ 表示允许该用户占用超过其比例的缓存空间(同时也表示了占用系统传输时间的公平性); 当信道条件较差时 $\alpha < 1$ ,  $\text{QLen}_{\text{ref}} \cdot \text{weight}_i \cdot \alpha$ 表示一定程度上限制该用户的分组进入缓存。

理想情况下, 假设对用户信道条件进行了有效的监控,  $\alpha$ 取为当前信道速率与平均速率的比值。同时考虑到对信道条件较差的用户性能的影响, 我们限制 $\alpha$ 取值在[0.5, 1]之间, 从而使得SI在两种公平性之间取得折衷, 即同时兼顾信道条件和分组丢弃的公平性。

## 2.3 模糊公平缓存管理算法FF-RED

分享指数的定义只是近似表征不同业务流对网络资源占用的相对大小, 仅使用分享指数作为拥塞判定显然是不适合的。因此我们提出使用分享指数与传统的缓存队列长度相结合的方式共同决定丢弃概率。考虑到系统的不确定性, 我们为分享指数和队列长度制定模糊规则, 采用模糊逻辑来进行拥塞控制。

在 FF-RED 算法中,输入语言变缓存队列长度  $Q$  和分享指数 SI 的语义分别解释为{短, 中, 长}和{负, 零, 正},即模糊化的项集合分别为  $T(Q)=\{Short, Medium, Long\}$ ,  $T(SI)=\{Negative, Zero, Positive\}$ , 输出语言变量丢弃概率 pdp 的语义为{低, 中, 高},即项集合为  $T(pdp)=\{Low, Medium, High\}$ 。我们对不同优先级的队列制定不同的模糊规则。图 2、图 3 给出了不同优先级队列中这 3 个变量隶属度函数的形式。

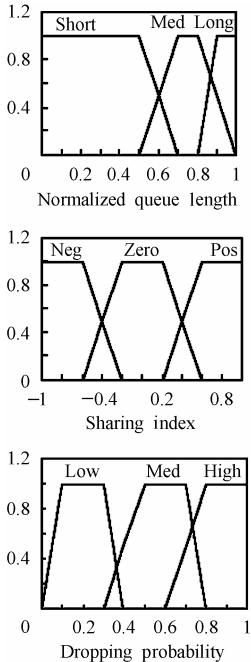


图 2 高优先级业务(In)的隶属度函数

Fig.2 Membership functions of high priority traffic (In)

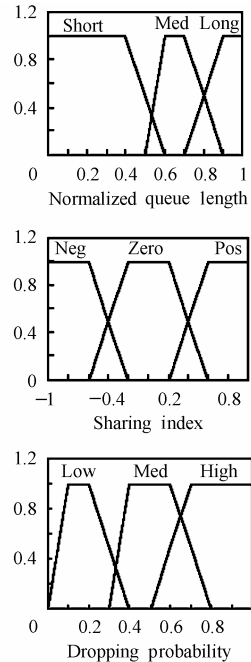


图 3 低优先级业务(Out)的隶属度函数

Fig.3 Membership function of high priority traffic (out)

FF-RED 算法在缓存队列达到一定长度时,可根据不同业务流对网络资源的占用情况(超过其分配比例还是小于分配比例,通过分享指数描述)以及队列的优先级进行处理,推断出以怎样的概率来丢弃后继的分组,从而克服了 RED 算法仅根据队列长度和固定门限值进行随机丢弃的缺点。

虽然文献中出现一些使用神经网络、遗传算法等方面知识制定模糊规则的方法<sup>[10]</sup>,但考虑到其复杂性,我们仍采用文献[11,12]中使用的较为传统的方法,即利用人的经验制定模糊规则,通过对仿真过程的观察,不断地对模糊规则进行调整。模糊规则描述如下:

```

/* linguistic rules for assured class */
if Q is short then pdp is zero;
if Q is medium and SI is negative then pdp is zero;
if Q is medium and SI is zero then pdp is zero;
if Q is medium and SI is positive then pdp is low;
if Q is long and SI is negative then pdp is low;
if Q is long and SI is zero then pdp is medium;
if Q is long and SI is positive then pdp is high;
/* linguistic rules for best-effort class */

```

```

if Q is short then pdp is zero;
if Q is medium and SI is negative then pdp is zero;
if Q is medium and SI is zero then pdp is low;
if Q is medium and SI is big then pdp is medium
if Q is long and SI is negative then pdp is low;
if Q is long and SI is zero then pdp is medium;
if Q is long and SI is positive then pdp is high;

```

### 3 仿真结果

#### 3.1 吞吐量和缓存队列长度的比较

我们使用 NS-2(版本 2.27)建立如图 4 所示的哑铃状网络结构,对缓存管理算法的吞吐量和时延进行比较。路由器 1、2 间的链路为网络瓶颈,带宽为 10Mbps,缓存空间大小为 30 个分组,假设所有分组具有相同的大小 1000 byte。瓶颈链路由 100 个用户共享,所有用户具有相同的传输时延 20ms。最初网络中没有活动业务流,每隔 1s 有一个用户开始发送数据,直到 100s 后所有用户进入活动状态。其中奇数秒进入网络的业务流为采用 TCPNewReno 的 FTP 连接,窗口大小为 32;偶数秒进入网络的为 Pareto 分布的 ON-OFF 业务源,On 期(burs ttime)和 Off 期(idle time)均为 500ms,速率为 200kbps, Pareto 分布参数  $\alpha$ (shape)为 1.5。

图 5,图 6 给出了 FF-RED 算法与 RED 算法有效吞吐量的比较曲线,图 7,图 8 给出了两种算法平均队列长度的比较曲线。其中 RED 算法使用 NS-2 的缺省设置。

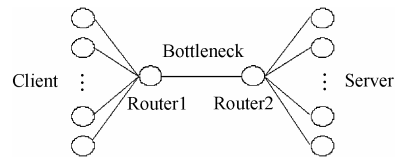


图 4 仿真网络结构

Fig.4 Network architecture in simulation

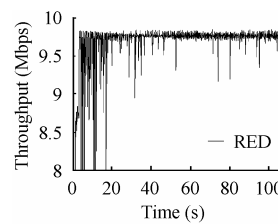


图 5 RED 算法有效吞吐量  
Fig.5 Throughput of RED algorithm

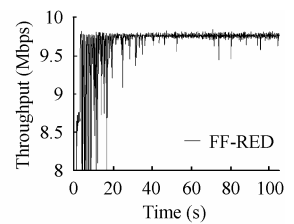


图 6 FF-RED 算法有效吞吐量  
Fig.6 Throughput of FF-RED algorithm

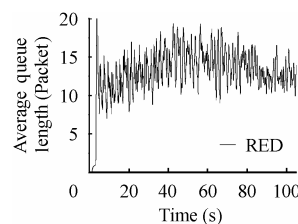


图 7 RED 算法平均排队长度  
Fig.7 Average gueue leugh of RED algorithm

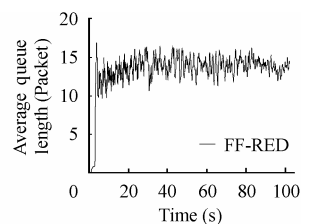


图 8 FF-RED 算法平均排队长度  
Fig.8 Average gueue Lleugh of FF-RED algorithm

一般来说，考虑公平性的算法会造成吞吐量的减小，图 6 中 FF-RED 算法的吞吐量曲线与 RED 算法的吞吐量比较并没有明显的性能上的损失。图 7，图 8 中的平均队列长度比较中，FF-RED 算法的平均队列长度明显较 RED 算法更为平稳(平均队长采用 EWMA 计算， $\omega=0.002$ 。由于两图均在较大的时间范围内绘制，因此平均队长的波动显得比较剧烈，如果仅取其中的 5s 对缓存队列长度和平均队长进行比较，FF-RED 算法的时延平稳性将更为明显)。我们认为主要是由于以下两个原因造成的：

(1)RED算法的设计目标是将平均队列长度控制在 $[\min_{th}, \max_{th}]$ 之间，而FF-RED算法在设计时兼顾了将分享指数SI保持在 0 附近的要求，即将每个业务流占用的缓存队列长度控制在 $QLen_{ref} \cdot weight_i$ 附近，从而将缓存队列长度控制在 $QLen_{ref}$ 附近，所以FF-RED算法的平均队列长度与RED算法相比显得更为平稳，即算法具有更平滑的时延特性。

(2)FF-RED 在进行拥塞控制时尽量使不同业务流具有相同的速率(权值相等时)。随机丢弃时，不论丢弃哪个业务流的分组，对拥塞的缓解程度都是比较接近的。因此缓存队列的变化受随机性的影响较小。

3.2 公平性的体现——对非响应流抢占带宽的抑制

当响应流占绝大多数时，对于 Internet 的正常运行和多用户共享带宽，TCP 的这种端到端拥塞控制机制是行之有效的。同时，RED 算法认为不同业务流的丢弃分组数与其速率成正比，速率越高被丢弃的分组数也较多，因此这种隐含的公平性要求能在一定程度上为不同业务流提供近似大小的有效吞吐量。

然而 RED 的这种公平性是非常脆弱的，随着较多非响应流(如以 UDP 为传输层协议的 IP 电话等)的加入，丢弃分组数与速率成正比这一假设不再成立。其后果是，有限的带宽被非响应流大量抢占，系统的公平性无法得到保证。这也是我们在计算分享指数时未采用计算更为简洁的对丢弃历史计数的方法的主要原因。考虑公平性的缓存管理算法，如 BRED、FRED 等算法对非响应流有一定的隔离作用，可防止其抢占网络资源，提供对响应流的保护功能。

为了比较两种算法的公平性，我们同样建立图 4 的模型。假设在 1s 时网络中同时有 4 个采用 TCPNewReno 的 FTP 连接。6s 和 8s 时分别有一个速率为 5Mbps 的 CBR 业务进入网络，图 9 给出了两种算法中 4 条 TCP 业务流总的吞吐量的比较。通过吞吐量随时间的变化，可以看到 TCP 与 UDP 两类业务流之间带宽的分配情况：当存在非响应流时，TCP 业务流的带宽会被非响应流抢占，最终 4 条 TCP 流的吞吐量会趋于 0，即 UDP 与 TCP 占用带宽的比值为 10: 0，这个时候显然是不公平的；而采用 FF-RED 算法后，4 条 TCP 流在非响应流进入网络后，其吞吐量会逐渐恢复到公平分配的比例，算法对非响应流抢占带宽的抑制作用是非常明显的。

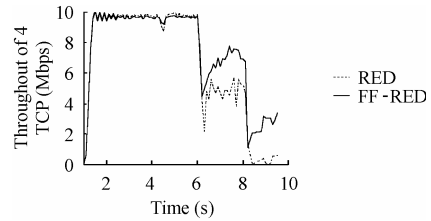


图 9 对非响应流抢占带宽的抑制

Fig.9 Restrain non-response flows occupy buffer excessively

3.3 无线网络中系统吞吐量的比较

在无线网络仿真中，我们对图 4 的仿真结构模型略微进行修改，客户端改为移动用户，此时用户与路由器 1(或者基站)之间的链路条件随用户所在位置和干扰情况而发生变化，即带宽随时间变化的无线链路。同时路由器 1 与路由器 2 之间的链路带宽足够大，网络瓶颈从有线部分转移到无线网络中。

为了简化无线信道的仿真，我们通过 32 态马尔可夫链来模拟每个移动中的用户所对应的无线信道，每个状态对应一个干扰指数，不同的干扰指数决定了信道的传输速率。状态 $Y_0$ 对应最优的信道条件，具有最高的传输速率，状态 $N$ 对应最差信道的条件。状态转移仅限于相邻状态之间，转移概率与用户的移动方向有关。用户的传输速率受到信道条件的限制。状态转移图及转移概率如图 10 所示。

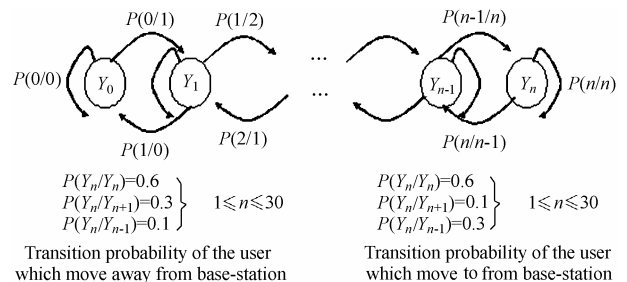


图 10 马尔可夫状态转移图与状态转移概率

Fig.10 Markov chain model of channel condition and transition probability

其中状态 $Y_0$ 和状态 $Y_{31}$ 为弹性壁状态，其转移概率为 $P(0/0)=0.6$ ， $P(0/1)=0.4$ ， $P(31/31)=0.6$ ， $P(31/30)=0.4$ 。

图 11 给出了无线网络中 RED 算法与 FF-RED 算法吞吐量与缓存队列长度的比较曲线(业务流数固定为 20)。由于在式(5)的 SI 定义中，我们对公平性的含义进行推广，允许信道条件较好的用户占用较多的缓存空间，即优先允许信道条

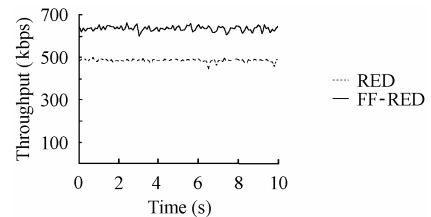


图 11 无线网络中两种算法吞吐量比较

Fig.11 Throughput comparison in wireless network

件较好的用户分组进入缓存。因此图 11 中采用 FF-RED 算法的吞吐量性能高于未考虑信道条件的 RED 算法的性能。

#### 4 结束语

本文提出了一种基于模糊逻辑控制的缓存管理算法——FF-RED。我们引入一个新的变量——分享指数，取代缓存占用量来表征不同业务流对网络资源的占用情况，以克服缓存占用量表征公平性的缺陷。同时，考虑到无线网络的一些特点，对无线网络中分享指数的定义进行推广，使其同时兼顾公平性和系统性能(信道条件)。FF-RED 算法通过对分享指数和缓存队列长度共同制定模糊规则，使得在根据队列长度计算丢弃概率时，可根据不同流对网络资源的占用情况进一步决定增加或减小相应的丢弃概率，从而可利用不同流之间的差异以加速缓解拥塞，具有更好的自适应性和公平性。仿真表明，算法在满足高吞吐量、低时延要求的同时，在适应网络负荷变化、防止非响应流抢占带宽等方面具有较好的性能。

#### 参考文献

- [1] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. on Networking*, 1993 (1): 397-413.
  - [2] Clark D, Fang W. Explicit allocation of best effort packet delivery service. *IEEE/ACM Trans. on Networking*, 1998, 6(4): 362-373.
  - [3] Anjum F M, Tassiulas L. Fair bandwidth sharing among adaptive and non-adaptive flows in the Internet. Proc. INFOCOM'99, New York, USA, 1999: 1412-1420.
  - [4] Lin D, Morris R. Dynamics of random early detection. Proc. ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Cannes, France, 1997: 127-138.
  - [5] Feng W, Kandlur D, Saha D, Shin K. A self-configuring RED gateway. Proc. INFOCOM '99, New York, USA, 1999: 1320-1328.
  - [6] Ott Teunis J, Lakshman T V, Wong Larry H. Stabilized-RED. Proc. IEEE INFOCOM'99, New York, USA, 1999: 1346-1355.
  - [7] Feng W, Kandlur D, Saha D, *et al.* BLUE: a new class of active queue management algorithm. Proc. INFOCOM'2001, Alaska, USA, 2001: 1520-1529.
  - [8] Hollot C, Misra V, Towsley D, Gong W B. On designing improved controllers for AQM routers supporting TCP flows. Proc. INFOCOM'2001, Alaska, USA, 2001: 1726-1734.
  - [9] Athuraliya S, Li V H, Low S H. REM: active queue management. *IEEE Network*, 2001, 15(3): 48-53.
  - [10] Fatta G D. A genetic algorithm for the design of a fuzzy controller for active queue management. *IEEE Trans. on System, Man, and Cybernetic*, 2003, 33(3): 313-324.
  - [11] Chrysostomou C, Pitsillides A, Rossides L, *et al.* Congestion control in differentiated services networks using Fuzzy-RED. *Elsevier Control Engineering Practice*, 2003(11): 1153-1170.
  - [12] Fan Yanfei, Ren Fengyuan, Lin Chuang. Design an active queue management algorithm based fuzzy logic decision. Proceedings International Conference on Communication Technology 2003, Beijing, China, 2003: 286-289.
  - [13] Wang Chonggang, Li Bo, Sohraby Kazem, *et al.* AFRED: an adaptive fuzzy-based control algorithm for active queue management. Proc. IEEE Local Computer Networks 2003, Bonn, Germany, 2003: 12-20.
- 陈 远: 男, 1975 年生, 博士生, 研究方向为 IP 网络中的 QoS 和无线资源管理。
- 李乐民: 男, 1932 年生, 教授, 博士生导师, 中国工程院院士, 研究方向为通信网与宽带通信技术。