

## 无线网络中实时业务的随机超时早检测缓存管理算法

陈远 李乐民

(电子科技大学通信与信息工程学院 成都 610054)

**摘要** 该文提出一种适用于无线网络中实时业务的随机超时早检测缓存管理算法。考虑到无线网络具有动态变化的信道条件和存在传输错误等特点,算法分为3个层次:以保证重传分组不超时为目标,采用类似于拥塞早检测的方法对超时进行早期检测和随机丢弃;根据信道条件的变化趋势对丢弃概率进行自适应调节;根据信道速率的变化对门限值进行调整。在算法设计时,同时兼顾了分组丢弃对TCP性能的影响。仿真表明,采用该算法后,系统性能得到改善。

**关键词** 无线网络, 缓存管理, 服务质量, 信道条件, TCP

中图分类号: TN915.07

文献标识码: A

文章编号: 1009-5896(2006)05-0769-05

## A Random Early Expiration Detection Based Buffer Management Algorithm for Real-Time Traffic over Wireless Networks

Chen Yuan Li Le-min

(School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China)

**Abstract** A random early expiration detection based buffer management algorithm is proposed for real-time traffic over wireless networks. Considering the characters of wireless link such as variable channel condition and transmission error, the algorithm is composed of three steps: aimed to guarantee the delay of retransmission packets, using the method which is similar to early congestion detection to detect expiration and discard packets in advance; adaptive adjusting the drop probability based on the trend of channel condition; re-calculating the threshold value when the link rate is changed. The influence on the TCP performance is also considered in the design. Simulation results show that the system performance is improved through using the proposed algorithm.

**Key words** Wireless network, Buffer management, QoS (Quality of Service), Channel condition, TCP

### 1 引言

随机早期检测<sup>[1]</sup>(Random Early Detection, RED)是目前研究得最多的一类主动队列管理技术。它采用概率判定机制主动地有选择地丢弃某些分组,利用TCP对发送速率的自适应调节能力,让某些源端降低发送速率,及时阻止拥塞的恶化,并将平均排队时延控制在一定的范围内。RED采用指数加权滑动平均(EWMA)来计算平均队长,由平均队长决定分组丢弃概率。当流的数目较多时,很难设置合适的参数将平均队长维持在高低门限之间。Adaptive-RED<sup>[2]</sup>是一种典型的自适应RED改进算法,它根据网络拥塞程度对丢弃概率 $\max_p$ 进行自适应调整:当拥塞程度增大时,最大丢弃概率增大为 $\beta \max_p$ ;当拥塞程度减小时,最大丢弃概率减小为 $\max_p/\alpha$ 。文献[3]对常用的缓存管理算法进行了详细地分析和总结。

基于分组业务的无线网络中同样需要对缓存进行有效的管理。然而无线网络有其自身的特点,直接应用有线网络中成熟的缓存管理算法可能存在问题。在无线网络中会遇到如下情况:(1)无线链路的速率通常小于光纤链路,因

此同样的缓存大小,无线网络的排队时延要高于光纤网络中的排队时延。(2)无线链路存在传输错误,同时从功率控制的角度,维持一定的错误概率也是提高系统容量的必要手段。当发生传输错误时,一般采用前向纠错(FEC)进行恢复或者通过自动重传(ARQ)重新传输。因此,缓存管理算法需保证重传分组的时延也在允许的最大时延范围之内。(3)无线信道条件随时间动态变化,基站可根据网络负载和用户信道条件对用户速率进行动态调整。在小区边缘的用户可能分配较小的速率,而在基站附近的用户可获得较大的速率。显然,如果这两类用户的业务流具有相同的时延限制,所需维护的缓存队列长度是不同的。同时,随着用户的移动,同一用户不同时刻的速率也可能不一样,因此,缓存管理算法需具有对信道速率变化的自适应性。

文献[4]对RED在无线网络中的应用进行了分析和研究,认为直接使用RED算法存在一些问题。文献[5]使用RTS(Request-To-Send)的重传次数与时延之积作为拥塞指示,取代RED中的平均队长进行丢弃概率的计算,主要用于存在信道竞争和碰撞时无线局域网中的缓存管理。文献[6]同样采用类似于RED的算法,以时延限制计算丢弃门限,强调了重传对缓存管理的影响,然而并没有考虑信道条件的变化,也没有考虑分组丢弃对TCP性能的影响。

2004-09-13收到,2005-03-10改回  
国家自然科学基金委员会和香港研究资助局联合科研基金(60218002)资助课题

文献[7]认为在多速率的情况下,通过检测信道条件,优先丢弃具有最长传输时间的分组将能够达到有效利用无线资源的目的。文献[8]进一步提出在丢弃最差信道条件分组的同时还应该考虑公平性,并通过自适应参数在系统性能和公平性之间取得平衡。文献[4,9]研究了多速率情况下缓存管理算法对TCP性能的影响,并给出了理想的缓存队列长度。文献[10]认为时延中隐含了缓存队列长度、信道条件等信息,提出以维持平稳的时延为目标,采用类似于有线网络中基于比例积分器(Proportional-Integral, PI)的缓存管理算法。在上述考虑信道条件变化的算法中都没有考虑传输错误和重传对缓存管理算法的影响。

本文提出一种适用于无线网络中实时业务的随机超时早检测缓存管理算法,其基本思想在于缓存接纳新到分组时,对其能否在规定的时延限制之内到达收端进行估计,并根据估计结果进行随机早期丢弃,防止无效的传输占用宝贵的无线资源,同时也有利于减小排队时延,降低后继分组超时的概率。考虑到无线网络具有动态变化的信道条件和存在传输错误等特点,我们对有线网络中常用的自适应RED算法进行扩展。

## 2 系统模型与算法描述

图1给出了3GPP制订的UMTS规范中RLC层错误恢复与重传机制的简化示意图<sup>[6]</sup>。RLC采用基于窗口的选择性ARQ协议进行错误恢复。发端持续发送其传输队列中的分组(或协议数据单元PDUs,下文统一简称为分组),直到未确认的发送分组数到达窗口上限。发端在发送分组的同时,将刚发送分组保存在重传队列中。RLC根据收端返回的状态信息决定后继的操作:当收到ACK确认信息后,分组将从重传队列中删除;当收到收端未成功接受分组或接受到的分组发生错误时返回的NACK信息时,相应的分组将复制到传输队列中,以高于其它分组的优先级进行重传。当同一分组重传 $N$ 次( $N$ 由系统预先定义)后,收端仍不能正确接收,RLC将丢弃该分组<sup>[11]</sup>。

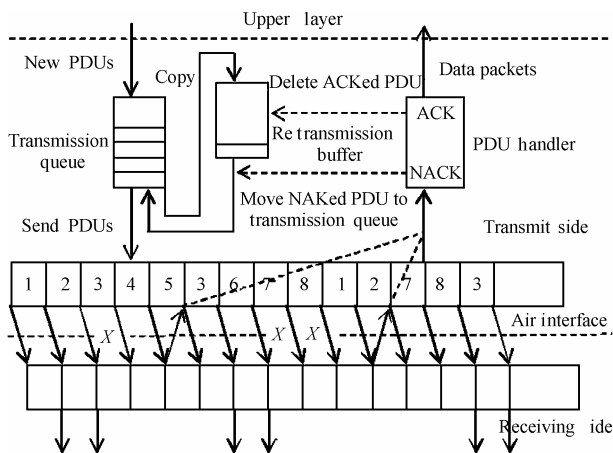


图1 UMTS中的无线链路控制层RLC(窗口大小为8)

Fig.1 Radio Link Control (RLC) layer of UMTS

对于实时业务来说,一般都有严格的时延限制,如果分

组到达收端时已超过其允许的最大时延,分组将被丢弃。在无线网络中,如果无线信道有较大的误码率,分组会在传输队列中不断累积,造成队列长度的不断增加,从而使排队时延增大,收端收到超过最大时延限制分组的概率也相应增大。因此我们希望有一种超时的早期检测机制,可以根据队列长度和信道条件估计进入缓存队列的分组到达收端时超时的可能性,并采取相应的早期丢弃策略,防止无效的分组传输占用宝贵的无线网络资源,同时在保证吞吐量的前提下,减少有效传输分组的排队时延,降低网络拥塞的可能性。

我们提出的早期检测随机丢弃的策略类似于拥塞控制中的RED算法,称之为自适应随机超时早检测(Adaptive Random Expiration Early Detection, A-REED)算法。从本质上看,RED算法和A-REED算法都是缓存管理算法,其目标都是与流控机制(如TCP)相结合,提供较大的吞吐量和较小的排队时延。然而无线网络中较大的重传概率和时变的传输速率使得RED算法中固定门限很难适应无线网络环境的动态变化。A-REED算法可看作以超时重传为限制确定门限、随信道条件进行自适应调整的RED算法,它通过超时早期检测合理有效地分配无线网络资源,利用早期丢弃无效分组来缓解拥塞、减小排队时延,是RED算法在无线网络中的扩展和延伸。

A-REED算法分为以下3个主要步骤:利用超时早期检测对可能超时的到达分组进行随机丢弃;根据信道条件的变化趋势对丢弃概率进行调节;根据信道速率的变化对门限值进行调整。

### 2.1 REED 随机超时早检测

我们采用类似文献[6]中的方法进行分析(式(1)~式(4))<sup>[6]</sup>。假设某一时刻某一用户的链路速率为 $R$ ,传输分组具有固定长度 $l$ ,则传输一个分组需要 $R/l$ 秒。如果每个分组只传输一次,收端就能正确接收,下式给出了队列长度的上限:

$$L_{\max} = \left\lfloor \frac{(T - t_1 - t_2)R}{l} \right\rfloor \quad (1)$$

其中 $T$ 为实时业务所允许的最大时延限制, $t_1$ 为发端到基站的网络总时延, $t_2$ 为基站到用户的无线传输时延及基站处理/调度时延的总和。显然,当一个新的分组到达时,如果缓存队列长度超过 $L_{\max}$ ,表示即使排队分组都能一次传输成功,新到分组到达收端时也会因为超时而被丢弃。因此队列中 $L_{\max}$ 之后的分组是没有意义的。

在最糟糕的情况下,假设每个分组都需要传输 $N$ 次才能正确接受(或者超过重传限制而被丢弃),则每个分组总传输时间为

$$t_3 = N \frac{l}{R} \quad (2)$$

同时分组在重传队列中等待ACK或者NACK的总时间为

$$t_4 = (N-1) \cdot t_2 \cdot 2 \quad (3)$$

此时可得到队列长度的下限为

$$L_{\min} = \left\lceil \frac{T - t_1 - t_2 - t_3 - t_4}{t_3} \right\rceil \quad (4)$$

显然, 当队列长度小于  $L_{\min}$  时, 即使在最恶劣的情况下, 分组仍能保证到达收端时延在允许的范围之内。

式(1)和式(4)给出了极端情况下队列的上下限, 我们从概率的角度考察队列最有可能的临界长度。假设发生传输错误的概率为  $\rho$ , 且错误的发生相互独立, 当前队列中有  $L$  个分组, 则其中的  $(1-\rho)L$  个分组能一次传输成功,  $\rho(1-\rho)L$  个分组需要两次, 依次类推可得  $\rho^{N-1}(1-\rho)L$  个分组需要传输  $N$  次。因此队列长度  $L$  需要满足下式:

$$(1-\rho)L(t_1+t_2) + \sum_{i=2}^N \rho^{i-1}(1-\rho)L \left[ t_1+t_2+i\frac{L}{R}+(i-1)t_2 \cdot 2 \right] \leq T \quad (5)$$

因此得到一般情况下队列处于超时与不超时之间的临界长度:

$$L_{\text{med}} = \left\lceil \frac{T}{(1-\rho)(t_1+t_2) + \sum_{i=2}^N \rho^{i-1}(1-\rho) \left[ t_1+t_2+i\frac{L}{R}+(i-1)t_2 \cdot 2 \right]} \right\rceil \quad (6)$$

$N$  一般取比较小的值, 因此上式的计算并不复杂, 且考虑到队列长度和  $\rho$  的实际取值, 重传 3 次以上对式(6)的影响很小, 计算时可忽略不计。

通过上面的分析, 当队列长度介于  $[L_{\min}, L_{\max}]$  之间时, 如果以一定的概率进行预丢弃, 可以将队列长度维持在允许的范围之内以防止分组到达收端时超时, 从而提前放弃无效传输以提高无线资源的利用率。当队列长度大于  $L_{\min}$  且更接近  $L_{\min}$  时, 以较小的概率进行丢弃; 而当队列长度接近  $L_{\max}$  时, 即使所有分组都能以较大的概率一次传输成功仍有可能在到达收端时超时, 因此此时采用较大的丢弃概率。这种预丢弃的方式非常类似于缓存管理中的 RED 算法。图 2 中用虚线给出了一种可行的超时早期检测的丢弃概率曲线。

虽然这条丢弃概率曲线符合我们前面的分析, 对超时早期检测有一定的作用, 但是从缓存管理的角度来看, 该方法

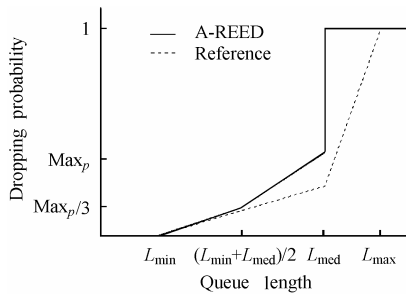


图 2 A-REED 丢弃概率曲线  
Fig.2 Dropping probability of A-REED algorithm

存在两个问题: 一是该方法将队列长度维持在超时与不超时的临界点, 不利于减小排队时延, 也不利于算法的适应性(例

如当分组丢失的发生不是相互独立而是由于突发链路错误时, 后继到达的分组可能会全部超时); 二是当队列长度接近  $L_{\max}$  时, 会发生缓存连续丢弃新到分组的现象, 这对一些具有流量控制的业务流(TCP)是非常不利的, 会在短时间内造成速率的急剧下降。因此我们采用图 2 中实线所示的略高于第 1 种方法的丢弃概率, 以  $L_{\text{med}}$  为缓存管理算法的上限, 在大于  $L_{\text{med}}$  时实行全丢弃策略(虽然此时在接近  $L_{\max}$  时仍然会出现连续丢弃的情况, 但“接近  $L_{\max}$ ”这一事件本身发生的概率远小于参考曲线)。丢弃概率的计算满足下式:

$$P = \begin{cases} 0, & n \leq L_{\min} \\ \frac{\max_p}{3} \frac{n - L_{\min}}{L_{\min} + L_{\text{med}} - L_{\min}}, & L_{\min} < n \leq \frac{L_{\min} + L_{\text{med}}}{2} \\ \frac{\max_p}{3} + \frac{2 \max_p}{3} \frac{n - \frac{L_{\min} + L_{\text{med}}}{2}}{L_{\text{med}} - \frac{L_{\min} + L_{\text{med}}}{2}}, & \frac{L_{\min} + L_{\text{med}}}{2} < n < L_{\text{med}} \\ 1, & n \geq L_{\text{med}} \end{cases} \quad (7)$$

## 2.2 丢弃概率 $\max_p$ 及其自适应调整

在 REED 中我们仅考虑了重传的影响, 而在无线网络中, 信道条件也会随时间发生动态变化, 在第 2, 第 3 步中我们将考虑如何在信道条件发生变化时对算法进行自适应调整。

假设某一时刻根据图 2 计算得到丢弃概率为  $P$ , 显然, 如果此时信道条件呈现转好的趋势, 以概率  $P$  丢弃可能丢掉一些本可以在允许时间范围之内到达收端的分组(即误丢弃); 而如果此时信道条件呈现变坏的趋势, 以概率  $P$  丢弃并不能很好地实现超时早期检测的目的。因此根据信道条件的变化趋势对丢弃概率进行微调是有必要的。实际上, 以信道条件的变化作为控制参数在无线网络中也是经常使用的, 例如高通公司的调度算法首先将业务流分成信道条件转好和变差的两个组, 再依据其它策略优先调度信道条件转好的一组中的业务流。

如何测定信道条件的变化在不同的系统中采用的方法略有不同。在我们的算法中采用一种简化的方法, 利用 3G 协议中定义的功率控制位(Power Control bit, PC bit)对丢弃概率进行微调。PC 位是基站发送给移动用户用于控制用户上行信道发射功率的一种方式。当基站接收到的用户信号功率不足以满足所需的 QoS 时, PC 位置 0, 用户接收后将相应的增大发射功率; 反之, PC 位置 1, 用户减小发射功率。假设上下行信道条件相同, PC=1 表示此时基站与用户之间的信道条件在逐渐变好, 干扰逐渐减小, 因此信道速率将有较大的概率增大<sup>[12]</sup>。功率控制的周期一般较短, 利用 PC 位对丢弃概率进行微调是可行的。

$$\left. \begin{aligned} \text{cum} &= \text{cum} + 1, & \text{PC} &= 0 \\ \text{cum} &= \text{cum} - 1, & \text{PC} &= 1 \end{aligned} \right\} \quad (8)$$

$$\max'_p = \begin{cases} \min[(1 + \text{cum}\Delta) \max_p, \beta \max_p], & \text{cum} > 0 \\ \max[\alpha \max_p, (1 + \text{cum}\Delta) \max_p], & \text{cum} \leq 0 \end{cases} \quad (9)$$

式(8)中cum为累积计数器,用于记录信道条件的总变化,式(9)中利用 $\alpha$ 和 $\beta$ 对微调范围进行限制,即 $\max_p \in [\alpha \max_p, \beta \max_p]$ , $\Delta$ 为微调变量,与调整时间间隔有关。调整间隔越短,表明对信道条件变化的测量越频繁,因此调整幅度可以小一些,对应着 $\Delta$ 可以取较小的值。

算法的参数设置随着无线网络通信情况的变化不断调整,这可能会影响整个系统性能的稳定性,在一定程度上限制该算法的使用。由于无线网络的复杂性,对算法稳定性的分析可能比较困难。一般可通过参数 $\alpha$ 、 $\beta$ 和 $\Delta$ 的调整对队列长度的振荡幅度进行限制。

### 2.3 门限值 $L_{\min}$ 、 $L_{\text{med}}$ 的自适应调整

在第2步中,PC=1表示信道速率将以较大的概率增大,但它仅仅表明了一种趋势,并没有真正意义上的速率变化。当这种趋势累积到一定程度时,基站会根据用户的信道条件和网络负载对速率进行调整,3G网络中速率可以达到384kbps,也可能低至8kbps。显然,对不同的速率维护不同的队列长度是不合理的。在第1步中我们给出的门限值也是与速率相关的。因此,此时需要根据调整后的速率对门限值进行重新计算。

当速率增大时,这种门限值的调整不会带来太大问题,然而在速率减小时,如果重新计算得到的高门限 $L_{\text{med}}$ 小于当前缓存队列长度 $L_{\text{current}}$ ,会造成分组的连续丢弃从而影响系统性能(尤其是TCP性能)。因此需要一种较为温和的策略将队列长度减小至新的高门限 $L_{\text{med}}$ 之下。令高门限值等于当前队列长度,我们采用计数器对到达分组进行计数,每到达 $m$ 个分组后门限值减1,直至门限值等于根据式(6)计算得到的目标值。算法的伪码描述如下:

```

 $L_{\text{target}} = L_{\text{med}}$ , based on 式(6)
 $L_{\text{med}} = L_{\text{current}}$ 
while  $L_{\text{med}} > L_{\text{target}}$ 
  count =  $m$ 
  while count > 0
    if new packet arrive
      count = count - 1
   $L_{\text{med}} = L_{\text{med}} - 1$ 

```

当速率变小时,最小门限 $L_{\min}$ 也相应减小,但缓存队列并不是越短越好。缓存管理算法需要维持一定的队列长度以取得吞吐量和排队时延之间的折中。文献[4]中认为,缓存队列长度至少要大于管道容量(Pipe capacity)才能有较好的带宽利用率,管道容量的计算基于公式 $\text{PipeCap} = \text{DataRate} \cdot \text{RTT}_{e2e}$ ,其中RTT为端到端环路延迟。我们用此公式计算有重传时的管道容量。从前面的分析可知,有重传的无线网络可近似看作速率介于 $[R/N, R]$ 之间的无差错网络,其平均速率 $\text{DataRate} = (1 - \rho)R$ ,其中 $\rho$ 为发生传输错误的概率。因此可计算得到提高吞吐量的最小队列长度为 $L_{\text{pipe}} = \text{PipeCap} / (1 - \rho)R / \text{RTT}_{e2e}$ 。我们对最小门限进行如下式所

示的修改:

$$L_{\min} = \max \left[ (1 - \rho) \frac{R}{l} \text{RTT}_{e2e}, L_{\min} \right], L_{\text{pipe}} < \frac{L_{\min} + L_{\text{med}}}{2} \quad (10)$$

通过对高、低门限的调整,我们以暂时、少量的降低带宽利用率为代价,避免了连续丢弃和短队列对系统整体性能的影响,提高了算法在速率发生变化时的适应性。

## 3 仿真结果及分析

我们建立如图3所示的系统模型对算法性能进行仿真分析。考虑到无线网络中基站一般为每个移动用户维护一个缓存队列,因此仿真中使用4条发往同一用户的业务流。为考察缓存管理算法与流量控制算法共同作用下的性能,其中两条为非响应的CBR业务流,另两条为承载实时业务的TCP流。假设分组具有相同的大小,且两类业务都有严格的时延限制,最大允许端到端时延为250ms。图3中服务器与基站间通过有线网络连接,传输延迟固定为50ms。基站与用户间为典型的无线链路,平均丢包率 $\rho$ 为15%,链路速率根据信道条件的不同在[8kbps, 384kbps]间取值。令RLC的最大重传次数为 $N=5$ ,算法的门限值可通过第2节的公式计算得到。

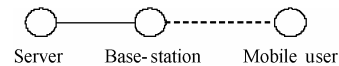


图3 仿真模型

Fig.3 Simulation model

为了简化仿真,我们预先定义用户的运动轨迹,假设用户沿直线运动,先移动至基站方向,再背离基站移动,其间经历了速率从128kbps至256kbps,256kbps至384kbps,384kbps至256kbps,256kbps至128kbps共4次的速率变化。 $\max_p$ 的调整间隔与功率控制周期相同,功率控制PC位的数值根据信道条件和功率变化相应的置1或0。仿真主要比较采用A-REED算法和未采用A-REED算法时系统性能的改变。未采用A-REED算法时我们采用标准的RED算法,考虑到仿真时有速率的变化,因此对RED算法进行修正。在速率发生变化时,对最大缓存队列长度进行调整。经过计算,RED算法在3种速率情况下的最大缓存队列长度分别设为30,60,90个分组。

图4,图5给出了两种情况下链路利用率和平均缓存队列长度的比较。链路利用率的计算为有效传输容量除以链路带宽,链路带宽根据时间的不同分别取为128kbps,256kbps和384kbps,而有效传输容量为扣除收端重复接收以及在收端因超时而丢弃的部分。缓存队列平均长度反映了系统的时延大小,采用EWMA进行计算,并选用较大的 $w$ 值以对长度变化做出较快反应(在A-REED算法中并没有平均队长的概念,此处只是对两种算法仿真结果进行计算引入的 $w$ 值,与仿真中RED算法采用的缺省参数不同)。

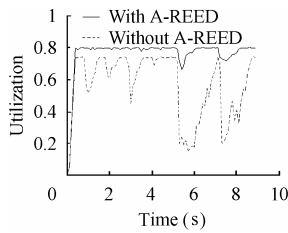


图4 采用/未采用 A-REED 时的吞吐量比较

Fig.4 Throughput comparison between with/without A-REED algorithm

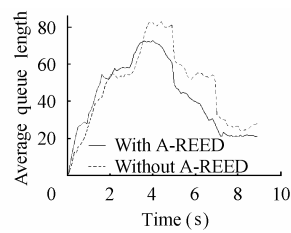


图5 采用/未采用 A-REED 时的队列长度比较

Fig.5 Average queue length comparison between with/without A-REED algorithm

从图4可以看到,即使选择合适的缓存队列长度,未采用 A-REED 算法的吞吐量与采用 A-REED 算法时比较仍然有一定的差距,而且选择合适的队列长度比较困难,也很难随信道条件的变化进行自适应调整。同时,未采用 A-REED 算法时系统吞吐量不如采用 A-REED 算法时平稳,图4中多次出现吞吐量的下降。这主要是因为分组传输到收端时因超时被丢弃造成带宽浪费,同时由于不能早期检测到超时,造成缓存队列长度增加,相应后继分组超时丢弃的概率也增加,连续的超时丢弃造成了链路利用率的下降。采用 A-REED 算法则可以尽量避免这种情况的出现。

在5s和7s,未采用 A-REED 算法的吞吐量出现了明显的下降,这主要是因为信道速率发生变化时对缓存队列进行调整未考虑分组丢弃对 TCP 性能的影响,造成源端发送速率的下降,而 A-REED 算法采用较为温和的方式调整门限(参见本文2.3节中的伪码),以浪费一小部分带宽为代价,换取 TCP 发送速率的相对稳定。A-REED 算法吞吐量下降不明显的另一个重要原因是我们在2.2节中通过 PC 位的变化对速率调整进行了预测,并对丢弃概率进行调节,在速率变化之前就已经通过式(8),式(9)将缓存队列长度逐渐减小,以削弱速率变化时对性能的影响。图5也可以看到这种缓存队列长度的自适应变化过程。

#### 4 结束语

本文提出一种适用于无线网络中实时业务的随机超时早检测缓存管理算法,其基本思想在于缓存接纳新到分组时,对其能否在规定的时延限制之内到达收端进行估计,并根据估计结果进行随机早期丢弃,防止无效的传输占用宝贵的无线资源,同时也有利于减小排队时延,降低后继分组超时的概率。考虑到无线网络具有动态变化的信道条件和存在传输错误等特点,我们对有线网络中常用的自适应 RED 算法进行扩展。在算法设计时,同时兼顾了分组丢弃对 TCP 性能的影响。仿真表明,采用该算法后,系统性能得到改善。

#### 参考文献

- [1] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1993, 1, (1): 397–413.
- [2] Feng W, Kandlur D, Saha D, et al. A self-configuring RED gateway. Proc. INFOCOM'99, New York, USA, 1999: 1320–1328.
- [3] 林闯, 单志广, 任丰原. 计算机网络的服务质量(QoS). 北京: 清华大学出版社, 2004: 147–167.
- [4] Sagfors Mats, Ludwig Reiner. Queue management for TCP traffic over 3G links. IEEE Wireless Communications and Networking Conference 2003, New Orleans, USA, 2003. 3: 1663–1668.
- [5] Dong Y, Makrakis D, Sullivan T. Network congestion control in Ad hoc IEEE 802.11 wireless LAN. IEEE Canadian Conference on Electrical and Computer Engineering 2003, Montreal, Canada, 2003. 3: 1667–1670.
- [6] Chen Jian, Leung Victor C M. Applying active queue management to link layer buffers for real-time traffic over third generation wireless networks. IEEE Wireless Communications and Networking Conference 2003, New Orleans, USA, 2003. 3: 1657–1662.
- [7] Seok Yongho, Park Jaewoo, Choi Yanghee. Queue management algorithm for multi-rate wireless local area networks. 14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, Beijing, China, 2003. 3: 2003–2008.
- [8] Chen Yuan, Li Lemin. A wireless packet dropping algorithm considering fairness and channel condition. International Conference on Communications Circuits and System 2004, Chengdu, China, 2004: 369–373.
- [9] Sagfors Mats, Ludwig Reiner. Buffer management for rate-varying 3G wireless links supporting TCP traffic. Vehicular Technology Conference 2003 Spring, Jeju, Korea, 2003. 1: 675–679.
- [10] Xu Heng, Xue Qi, Ganz A. Adaptive congestion control in infrastructure wireless LANs with bounded medium access delay. International Mobility and Wireless Access Workshop 2002, Texas, 2002: 44–49.
- [11] 3G TS 25.322. Radio Link Control (RLC) protocol specification. <http://www.3gpp.org>
- [12] Jeong Dong Geun, Jeon Wha Sook. Congestion control schemes for reverse link data transmission in multimedia CDMA systems. *IEEE Trans. on Vehicular Technology*, 2003, 52(6): 1489–1496.

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。