

一种提高异构网络传输性能的双向流量控制机制

郑明春^{1,2}, 杨寿保¹, 于晓梅², 孙伟峰¹

(1. 中国科技大学计算机科学系, 安徽合肥 230026 2. 山东师范大学计算机科学系, 山东济南 250014)

摘要: 针对异构网络存在的高误码率和不对称带宽等特点使 TCP 性能降低的问题, 本文提出了一种双向流量控制机制 BFC, 它根据数据链路的双向通信状况, 在前后向两个信道进行流量控制. 在前向信道, 我们设计了一种显式的 ACK 详细指示方法, 协调多层控制机制及时调节流量; 在后向信道, 提出了一个 ACK 流量控制算法, 根据双向拥塞状况, 对 ACK 流进行自适应地控制. 实验表明, 该机制可以有效地提高异构网络传输性能, 尤其是当双向信道的不对称参数较高的情况下, 该方法显示出独特的优势.

关键词: 流量控制; 无线网络; ACK; 不对称网络; SACK

中图分类号: TP391.1 **文献标识码:** A **文章编号:** 0372-2112 (2006) 05-0957-05

A Bidirection Flow Control Mechanism Improving the Transfer Performance of Wireless Network

ZHENG Ming-chun^{1,2}, YANG Shou-bao¹, YU Xiao-mei², SUN Wei-feng¹

(1. Department of Computer Science, University of Science and Technology of China, Hefei Anhui 230026 China;

2. Department of Computer Science Shandong Normal University, Jinan, Shandong 250014 China)

Abstract In order to solve the problem that performance of TCP is reduced due to higher bit error rate and bandwidth asymmetry in wireless networks we put forward a Bidirection Flow Control mechanism (BFC). It controls traffics in two directions. In the forward channel according an explicit ACK notification BFC adjusts data rate by cooperating multi-layer protocols. In the reverse one we developed an ACK flow control algorithm that adjusts self-adaptively ACK rate according to congestion status of two channels. The results of the simulation experiments show that BFC has dramatically improved the performance of TCP over wireless/wired hybrid links. Particularly, in the networks with higher asymmetric factor, BFC has significant contribution.

Key words flow control; wireless network; ACK; asymmetric network

1 引言

TCP/IP 的传输控制机制对于维持 Internet 的鲁棒性起了决定性的作用. 然而, 由于有线网络中链路的可靠性前提在无线网中并不成立, 因此, 现存的传输控制机制在无线网络中不能达到令人满意的性能. 国内外许多研究小组已经致力于无线网络传输控制技术的研究, 提出了一些改进机制, 但是目前依然没有公认的解决方案. 分析起来, TCP/IP 协议应用于含有无线链路的异构网络存在以下问题: (1) 信道误码率高使 TCP 拥塞避免算法可能产生错误的判断而导致吞吐量下降. (2) 信道的非对称性引起不必要的重传和数据突发现象. 因为无线网络大多采用非对称接入方式, 后向信道的传输速率往往比前向信道慢几倍到

几十倍^[1]. 这样, 后向信道的缓存队列很容易因过多的数据包排队而溢出, 导致拥塞丢包, 进而引发前向信道上大量数据包不必要的重传, 造成网络资源的浪费和 TCP 性能的下降. (3) 报文乱序问题. 近两年的研究表明, 无线网络数据包发生重排的比例达到 0.1% ~ 2%^[2], 数据重排现象可以引起 TCP 拥塞控制机制的误操作, 导致不必要的重传, 使 TCP 性能急剧降低.

为解决上述问题, 本文提出一种双向流量控制机制 BFC (Bidirection Flow Control).

2 相关工作分析及 BFC 思想的提出

针对上述 TCP 协议应用于无线网络存在的问题, 近年来许多研究者做了很多有意义的工作^[3-7]. 与本文相关的

有以下两个方面:一是关于不对称网络条件下流量控制机制的研究;二是关于区分无线网络的丢包原因的研究.以下分别简述这两方面的工作,并引出 BFC 机制的设计思想.

2.1 不对称网络流量控制研究

对于不对称带宽问题的研究,Internet草案 TcpSAT(Tcp over Satellite)^[1]综述了许多卫星通信网络中的技术,如延迟的 ACK,ACK 拥塞控制算法 ACC 和 ACK 过滤算法 AF 等.

为了叙述方便,以下我们约定前向信道为下行,后向信道为上行,约定发送者沿下行方向发送数据包至接收者,ACK 沿上行方向给发送者确认信息.

延迟的 ACK 方法是使一个 ACK 包中包含多个数据包的确认信息.这种方法可以有效地减少上行信道的拥塞程度,但是因为发送者根据它收到的 ACK 包中包含的确认包的数量来调节其拥塞窗口(cwnd)的大小,所以,延迟的 ACK 方法有可能导致发送者向网络注入突发的数据量,或使 cwnd 增长缓慢而导致网络性能下降.文献[3]详细统计分析了延迟的 ACK 方法影响 Web 流传输的情况,发现它能使 Web 流明显减少,特别是当拥塞窗口 cwnd 较小时.文献[1]也有相似的结论,它建议仅在拥塞避免阶段使用延迟的 ACK 方法.此外,文献[8]提出了一个改进的机制 ACE,让发送者将 cwnd 值作为一个新的 TCP 选项发送到接收端,接收端根据这个 cwnd 值计算下一个 ACK 包应该包含多少个数据包的确认(称为 ppa packets per acknowledgment).

综合以上分析可以看出:在慢启动阶段,不适宜使用延迟的 ACK,因为它会导致拥塞窗口增长缓慢;而在拥塞避免阶段,随着拥塞窗口的减少,相应地减少 ACK 包的发送频率,即增加 ACK 包中确认的数据包数,则会协调双向数据流量,提高网络性能,减少丢包率.从 ACE 算法和 ACC 算法中,我们还可以发现:一个 ACK 包确认的数据包数量,本文称为不对称因子 A(定义见 3.1 节),与上行信道拥塞程度成正比,而与下行信道拥塞程度成反比.再看 ACE 算法中 ppa 的计算:

$$\text{if}(\text{est_cwnd} \leq \text{min_ack_per_win}) \text{ then ppa} = 1 \\ \text{otherwise} \\ \text{ppa} = \text{MIN}(\text{FLOOR}(\text{est_cwnd}/\text{min_ack_per_win}), \\ \text{max_packets_per_ack})$$

其中, min_ack_per_win 为每个窗口发送的最少 ACK 包数,设为常数 3 而 max_packets_per_ack 为 ppa 最大值. ACE 只给出 ppa 与下行拥塞状况的关系,而没有给出 ppa 与上行拥塞状况的显式关系.为此我们想到,可以找出 ppa 与上行信道状况的关系,从而有效地控制 ACK 的发送频率,也就是后向流量控制.

2.2 无线网络丢包原因分析与处理

许多工作致力于无线网络环境下丢包问题的研究,如 Snoop 协议^[9]在基站处使用“软”状态探查代理,区分有线

拥塞丢包和无线传输错误引起的误码丢包.进一步,SY Wang 和 H T Kung 提出 TCP Decoupling^[10],建议使用差错控制与拥塞控制分离的算法,类似的还有 ELN^[11]方法等等.另一个与本文相关的工作是 S Floyd 等提出的 SACK 算法^[12],与 NEW Reno 方法相比, SACK 算法的最大优势在于它能够在一个 RTT 内重传多个丢失的数据包,由于该算法的 TCP 选项包含更多的信息,使源端可以更精确地估计传输路径上数据包的数量,而且实现了选择重传,因此,比 NEW Reno 更具鲁棒性.

以上工作说明,对于无线网络,明确区分丢包原因是十分必要的. BFC 的思想就是在明确丢包原因的前提下,协调多层控制机制,在双向链路上对流量进行有效地控制.为了实现这个目标, BFC 中的 ACK 包格式在 SACK 包的基础上进一步做了改进,这样,也有利于解决数据包的乱序问题.

3 双向流量控制机制 BFC

双向流量控制是指在前后两个方向进行流量控制, BFC 机制充分考虑了网络的双向通信状况,在前向信道,显式指示丢包原因和正确到达的情况,协调各层控制机制,对不同的情况,及时调节流量;在后向信道,估计双向拥塞状况,对 ACK 流进行自适应控制.仿真实验表明,与 SACK 和 Snoop 协议相比,对于高丢包率的无线网络环境,和不对称网络环境, BFC 明显提高了网络吞吐量,且没有引起数据突发问题.

BFC 机制的关键技术包括两部分:一是上行信道流量控制算法,二是多层协作的显式指示处理机制,以下重点介绍这两部分.

3.1 上行信道流量控制算法

在连接创建的三次握手阶段,发送端在发向接收端的 SYN 报文段中捎带发送一个 ASK 选项,请求使用新的算法进行数据传输.如果接收端返回 ANSWER 选项表示同意,则接收端可以使用 ESACK 选项报告数据包的接收情况.发送端与接收端分别执行以下操作:

- 发送端根据 cwnd 大小发送数据包,且捎带一个 A 选项,用以指示接收端发出的下一个 ACK 包应该包含 A 个确认的数据包;
- 接收端每收到 A 个数据包,就发送一个带有序列号的 ACK 包,且捎带一种新的 TCP 选项字段 ESACK,向发送端准确报告数据包的接收情况;
- 发送端收到来自接收端的 ACK 包时,检查每个 ESACK 块的标志位,看是否存在已标记的或丢失了的数据包,并相应调整拥塞窗口的大小.源端根据 ACK 包的接收情况判断后向信道上的链路状况,并计算下一个 A 选项字段的值.

算法中关键的参数 A 和 ESACK 格式讨论如下.

定义 1 不对称因子 (Asymmetric Factor) 源端发出

的数据包个数与接收端可以发送的 ACK 包个数之比. 用 A 表示. 计算公式是:

$$A = \text{Min} \left\{ \left\lceil \frac{awnd}{cack} \right\rceil, MaxA \right\} \quad (1)$$

其中 $cack$ 为每个窗口可以发送的 ACK 包的个数, $MaxA$ 为不对称因子最大值. 例如: A 为 4 表示每收到 4 个数据包, 数据接收端就发送 1 个 ACK 包确认收到的报文.

由 2.1 节分析, 当拥塞窗口较小时, 增大 A 会降低网络利用率, 所以, 我们设 $cack$ 的初值为 3 当 $awnd \leq cack$ 时, 令 $A = 1$ 那么如何根据上行信道的状况调整 A 的值呢? 首先讨论没有后向拥塞的情况. 算法取 A 是正整数且 $A \leq awnd$. A 为正整数保证了接收端对接收到的每个数据包能且最多只能发送一个 ACK 确认, 不允许对收到的同一个数据包发送两个以上的 ACK; $A \leq awnd$ 是为了保证同一个发送窗口的数据包至少能有一个 ACK 及时返回发送端, 以完成 RTT 值的正常估算, 并防止源端因无法及时收到 ACK 而导致重传计时器超时.

算法若没有检测到后向拥塞, 则令: $cack = \text{Min}((cack + 1), awnd)$. 那么 $cack$ 经过多少次加 1 A 才能减少 1 呢? 我们假设当后向信道连续 n 个 RTT 没有检测到拥塞, A 减 1, 即:

$$\lceil \frac{awnd}{cack + n} \rceil = A - 1 = \lceil \frac{awnd}{cack} \rceil - 1 \quad (2)$$

我们可以从公式 (2) 中推出 n 的值. 由上取整定义:

$$A - 1 < \frac{awnd}{cack} \leq A - 1 \text{ 得到 } n \geq \frac{awnd}{A - 1} - cack$$

所以, n 的最小值是:

$$\text{NT} \left\lceil \frac{awnd}{cack} - cack \right\rceil \text{ 其中 } (A > 1).$$

若检测到后向拥塞, 则令 $A = \text{Min}(A + 1, MaxA)$, 此时, A 的调整不使用公式 (1) 计算而直接线性加 1 是为了尽量减少发送端向网络中注入突发的数据包. 综上分析, 得到不对称因子 A 的算法如下:

```
初始时, cack = 3; MaxA = MAX(k, 1)
//k 为网络的归一化非对称参数
if (cwnd <= cack) A = 1;
else
{ A = Min(NT(cwnd/cack + 1), MaxA);
  n = NT(cwnd/(A-1) - cack);
  i = 0;
  WHILE (cwnd > cack) {
    if (没有后向拥塞) and (A > 2)
    { i = i + 1;
      cack = cack + 1;
      if (i = N A = A - 1; }
    else A = Min(A + 1, MaxA); } }
```

上述算法还有几个问题要说明: $MaxA$ 的选取、ACK 包选项块的格式以及如何检测后向拥塞.

$MaxA$ 的选取 在上述算法中, 选取 $MaxA$ 等于网络的归一化非对称参数 $k^{[6]}$. k 是衡量两条信道非对称性的主要参数, k 越大, 表明这两条信道对数据的处理能力相差越大. 一般地, 对称信道 $k \ll 1$ 对非对称链路, 后向信道带宽降低使得该信道对 ACK 包的处理速度减慢. 这时前、后信道的归一化带宽比 $k > 1$ 因此, 我们设定 $MaxA$ 为归一化非对称参数 k 在上述算法中当 $k < 1$ 时, $MaxA = 1$ 即回到原 TCP 的 ACK 确认机制.

ACK 选项块的格式 $MaxA$ 等于 k 的设定带来一个问题, 当 k 较大时, ACK 包不能容纳 k 个数据包的确认信息. 因此, 我们需要对 ACK 选项块进行改进. 如第 2 节所述, 为了减少各层对丢包的重复反应, 我们将使用显式指示反馈, ACK 包的格式在 SACK 算法的基础上进行改进, 我们将 ACK 选项块的格式称为显式 (Explicit) 指示的 SACK, 记为 ESACK, ESACK 的格式如表 1 所示.

表 1 ESACK 选项块格式

		种类 = 5	长度
本选项记录的数据段的终止序号			
块 1 状态	ACK 号	块 1 的持续长度	
块 2 状态	ACK 号	块 2 的持续长度	
st		con_len	
块 N 状态	ACK 号	块 N 的持续长度	

各字段代表的意义如下: TCP 选项长度和种类占去 2 个字节, 接下来的 4 个字节记录本次确认的数据段的终止序号, 若有用于 RTT 估计的时间戳选项则再占用 12 个字节 (其中有两个字节的填充), 这样余下的空间就用于记录目的端对各数据包的接收情况. 其中 Con_len 部分占 6 个比特位, 用于记录在目的端接收到的处于同一状态的连续数据段的长度, 最多可记录 63 个正确接收的数据段, 每一数据段的缺省值为 MSS (最大报文段长度) 个字节; ST 部分是接收情况标记位, 占 2 个比特, 其状态值反映了数据包的接收情况和链路状态, 具体含义见表 2 可见, 这里的每个 ESACK 块只需占用一个字节, 因此对于有 40 个可用字节的 TCP 选项来说, 最大可定义 34 个块, 即使是和时间戳选项一起使用, 也可定义 22 个块. 这与传统 SACK 最多有 3 个 SACK 块相比, 大大增加了可用块的数目, 即使在 k 较大的无线非对称链路中, 也可满足使用要求.

表 2 ESACK 状态值含义

状态值	代表的意义
00	数据包正确收到 (链路正常)
01	数据包带有 ECN 标记 (链路将拥塞)
10	只收到包首部 (损伤丢包)
11	没有收到该数据包 (其他丢包)

3.2 多层协作的显式指示处理机制

无线环境下丢包原因包括误码、拥塞和移动, 因此, 无论是链路层, 网络层还是传输层都无法仅仅依靠单层能力解决所有丢包问题, 各协议层应该相互协作才能更好地支

持端到端的可靠服务,得到不断优化和完善的 TCP 性能.我们结合协议各层的特点及优势,提出一种多层协作的显式丢包原因反馈机制.基本思想描述如下:

(1)当数据包到达路由器的数据链路层,调用该层的差错检验和控制机制对到来的数据帧进行校验并修复损伤的数据,对于损伤严重无法恢复的数据帧则丢弃数据部分,只把帧首部上传至高层.由于数据链路层能够检测出绝大多数受损的数据包,因此在传输链路中受损的数据要么得以修复,要么被丢弃,但绝大多数数据包的首部部分都能成功到达数据接收端.

(2)若数据包到达各路由器的网络层,则使用 ECN 算法监视该处数据包的排队情况:若平均队长超过丢弃水平就以规定的概率在其 IP 分组头中设置 CE 位一标记为“网络早期拥塞”(只对带有帧首部的包也是如此).

(3)对于到达目的端传输层的数据包,利用 ESACK 选项向源端反馈链路状况和数据传输情况,并显式反馈每个数据包丢失的具体原因.源端可以此为据,有针对性地调整数据发送率,并快速重传丢失的数据.

4 仿真分析

实验中,由于环境因素引起的无线损伤丢包现象是影响网络性能的重要因素之一,为真实反映不同传输协议在各种网络环境下的性能,我们使用指数分布的比特错误模型模拟无线环境下的包损伤现象:当出错产生器决定丢弃给定的数据包时,便按照指数分布以一定比率改变数据包 IP 首部的校验和.损伤丢包现象在无线链路的双向信道均会产生,因此对 ACK 而言,也存在同样的丢包概率.我们用两组实验测试 BFC 算法的性能.

第一组实验网络拓扑结构如图 1 所示,测试在高误码率的无线和有线混合网络中, BFC 算法的性能.

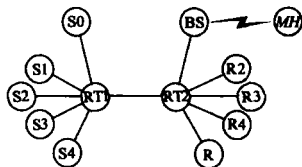


图 1 四跳网络拓扑结构

图 1 中,结点 S0~ S4 为发送端,移动主机 MH 及 R2~ R4 是目的端, BS 为基站, RT1 RT2 为瓶颈链路上的两个路由器.其中 S0→ MH 是观测流, S1→ R1 提供背景流.瓶颈链路参数见表 3 其他链路带宽均为 10Mbps 传输延迟为 2ms ESACK 选项块中的数据段以 500Bytes 为单位计算.

BFC 和 SACK 算法的拥塞窗口 (cwnd) 变化情况如图 2 所示,与 SACK 算法相比, BFC 算法的发送端在进入拥塞避免阶段后,其拥塞窗口 cwnd 振荡性较小,窗口变化平缓.这是因为 ESACK 选项包含的信息含量比 SACK 选项更丰富、更准确.它能够及时报告数据块的丢失情况,且显式标示报文丢失的具体原因,使源端可以较准确地判断链路状况,减少了无谓的窗口减半操作,从而减少了网络抖动,平均窗口大小也明显大于 SACK,可以获得更多吞吐量.

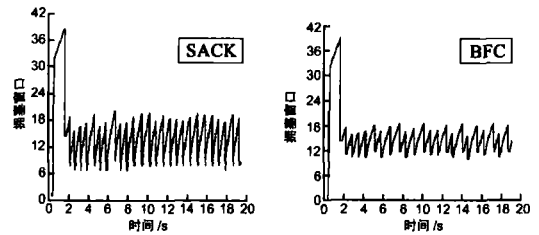


图 2 BFC 和 SACK 算法的拥塞窗口变化情况比较

表 3 瓶颈链路上主要实验参数

参数	参数值
链路容量	1.5M bps 10M bps 100M ps
前向链路的出错率 BER	$1.0 \times 10^{-9} \sim 1.0 \times 10^{-4}$
后向链路的出错率 BER	0或前后向 BER相同
单向传播时延	10ms 20ms 100ms 320ms
最大报文段长度 (MSS)	500~ 1000Bytes
接收方通告窗口	20~ 60segments
路由器缓存队列	20~ 50segments

以往的模拟实验结果表明,在无线网络环境下, Snoop协议表现出了优于 RENO 和 SACK 算法的网络性能^[4,7,9],是目前公认的无线网络中性能最好的基于窗口的传输控制机制之一.这里,我们将 BFC 算法嵌入 Snoop 协议中,形成 Snoop+ BFC,以下我们将比较 Snoop+ BFC 与 Snoop 协议、SACK 算法的性能差别.实验结果如图 3 所示, Snoop+ BFC 算法的网络性能最优, Snoop 协议次之, SACK 版本性能最差.

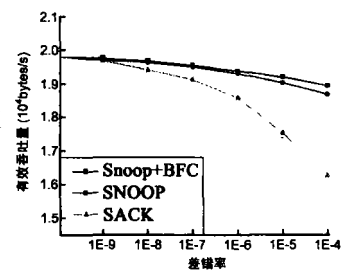


图 3 Snoop+ BFC 算法与 SNOOP 协议和 SACK 比较

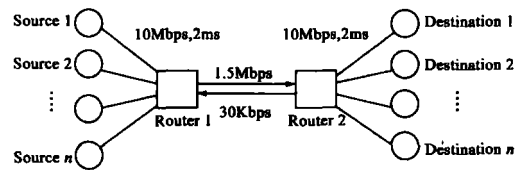


图 4 不对称链路实验拓扑结构

当差错率 BER 小于 10^{-6} 时, BFC 算法与 Snoop 协议的网络性能相当;当 BER 大于 10^{-6} 时,二者的网络性能开始出现较大差别, BFC 算法使网络利用率平均比 Snoop 协议提高 15%.

第二组试验的拓扑结构如图 4 所示,这组试验测试在不对称链路上 BFC 算法的性能.

我们用 5 个源端产生 Cross Traffic 数据流,本实验中 Cross Traffic 为 FTP 流,时间间隔设为 0.1 秒.我们分别调用 SACK 算法和 BFC 算法进行实验,从图 5 我们看到,若独立丢包率较低时(如 (a) 图 5%), BFC 算法与 SACK 算法

性能相当,而当丢包率
达到为 7% 时 ((b)
图),由于后向瓶颈链
路中有较多的 ACK 包
被丢弃,加剧了重传计
时器超时的概率,使用
SACK 算法时,数据发
送源端频繁进入慢启
动阶段,网络性能下
降,而 BFC 算法由于在
后向链路中加入了
ACK 拥塞控制策略,因
而比 SACK 算法获得了
更高的网络吞吐量.从
(c)图可以发现,当存
在突发丢包事件时,
BFC 显示出更明显的
优越性.这时 BFC 算
法使网络吞吐量大约
提高了 18%.

5 结论

针对现有异构网络
流量控制机制中的问
题,我们以提高单位
SACK 块的信息量为
切入点,设计了一种
新的带有显式标识的
双向流量控制机制
BFC 机制.实验表明,
BFC 机制在不对称
网络和高误码率异
构网络中,可以获得
更高的性能,而且
BFC 算法只修改了
40 字节的 TCP 选项,
因此该算法与传统的
基于窗口的 ACK 确
认机制不存在冲突.

参考文献:

[1] M Alman et al Ongoing TCP research related to satellite[S]. RFC 2760, 2000 2
[2] J C Bennett, C Partridge, N Schectman. Packet reordering is not pathological network behaviour [J]. IEEE/ACM Transactions on Networking, 2000, 7(6): 789-798
[3] R Cohen, S Rananathan. TCP for high performance in hybrid fiber coaxial broadband access networks [J]. IEEE Transaction on Networking, 1998, 6(1): 15-29
[4] S Fanglej, L Victor O K. Design of SNACK mechanism for wireless TCP with new snoop [A]. WCNC 2004 / IEEE Communications Society [C]. Atlanta: IEEE

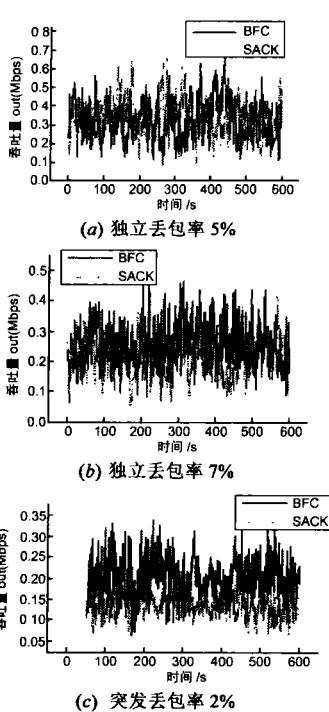


图 5 BFC 算法在不对称链路上的性能

Press, 2004, 1051-1056
[5] H Bakkrishnan, V N Padmanabhan, G Fairhurst. TCP performance implications of network path asymmetry [S]. RFC3449, 2002 12
[6] H Bakkrishnan, V N Padmanabhan, R H Katz. The effects of asymmetry on TCP performance [J]. ACM Mobile Networks and Applications, 1999, 4(3): 219-241.
[7] Sam a Vangala, Miguel A Labrador. The TCP SACK-aware snoop protocol for TCP over wireless networks [A]. In Proc of IEEE VTC 2003 [C]. Orlando: IEEE Press, 2003, 2624-2628
[8] IT Ming-Chit. Improving TCP performance over asymmetric networks [J]. ACM Computer Communication Review, 2000, 30(3): 45-54
[9] H Bakkrishnan. Challenges to reliable data transport over heterogeneous wireless networks [D]. University of California at Berkeley, USA, 1998
[10] S Y Wang, H T Kung. Use of TCP decoupling in improving TCP performance over wireless networks [J]. IEEE Mobile Networks and Applications, 2000, 27(6): 1-20
[11] H Bakkrishnan, R H Katz. Explicit loss notification and wireless Web performance [A]. Proc IEEE Globecom Internet Mini-Conference [C]. Sydney: IEEE Press, 1998, 260-278
[12] S Floyd, M Mahdavi, M Mathis et al. An Extension to the Selective Acknowledgment (SACK) Option for TCP [S]. RFC 2883, 1999 8

作者简介:



郑明春 女, 1963 年出生于山东省烟台市, 中国科技大学计算机系统结构专业在职博士生, 山东师范大学教授, 研究方向为计算机网络拥塞控制, E-mail: zhengm@ustc.edu.cn



杨寿保 男, 1947 年出生于上海市, 中国科技大学教授, 博士生导师, 中国科技大学网络信息中心主任, 研究方向为新一代互联网协议和移动计算等, E-mail: syang@ustc.edu.cn