

无线链路的TCP性能问题及其改善*

叶敏华** 刘雨 张惠民

(北京邮电大学信息工程学院 北京 100876)

【摘要】介绍了无线链路上的TCP性能问题和相关的工作。提出了一种改进从移动主机到固定网络方向TCP性能的方案,在基站处设置TCP代理来监视在无线链路上丢失的TCP报文段,通过复制和发送重复的ACK来触发MH端TCP的快速重传,有效地防止超时重传的发生。计算机仿真结果表明此方案具有良好的性能。

关键词 传送控制协议; 应答; 传送控制协议代理; 报文段

中图分类号 TN913.24 文献标识码 A

Improve the TCP Performance over Wireless Link

Ye Minhua Liu Yu Zhang Huimin

(School of Information Engineering, Beijing University of Posts and Telecommunications Beijing 100876)

Abstract The TCP performance on wireless link and the related works are introduced. A new scheme is proposed to improve the TCP performance over the wireless link on the direction from mobile host to fixed network. It installs a TCP agent at the base station to monitor the segments dropped on the wireless link, then triggers TCP fast retransmission on the MH side by replicating and forwarding specific ACKs, which can effectively avoid timeout retransmission. Simulation results show that the new scheme has favorable performance.

Key words transport control protocol; acknowledge; TCP agent; segment

1 TCP的拥塞控制机制与无线链路的特点

1.1 TCP中的拥塞控制机制^[1]

传送控制协议(Transport Control Protocol, TCP)是一种在固定网络中取得极大成功的传输层协议^[2],它使用累积的应答(Acknowledge, ACK)机制来判断哪些报文段(Segment)到达了接收方,并重传丢失的报文段来保证可靠传输。TCP维持两个变量:拥塞窗口cwnd和慢启动门限ssthresh。当cwnd小于ssthresh时, TCP处于慢启动阶段;当cwnd大于ssthresh时, TCP进入拥塞避免阶段。在发送过程中,若重传定时器超时, TCP判定报文段丢失:将ssthresh设为当前窗口的一半, cwnd置为1个报文段大小,重传丢失的报文段,进入慢启动阶段。若收到3个重复的ACK, TCP进行快速重传和快速恢复^[3]:

- 1) 设ssthresh为当前拥塞窗的一半, cwnd为ssthresh加3个报文段大小,重传丢失的报文段;
- 2) 每收到一个重复的ACK, cwnd增加1个报文段大小,如果cwnd允许,则发送一个报文段;
- 3) 当确认新数据的ACK到达时,恢复cwnd为ssthresh,退出快速恢复阶段,进入正常的通信阶段。

1.2 无线链路的特点导致TCP性能下降

无线通信将在未来的数据通信中起着重要的作用。无线链路的特点可归结为高误码率、低速率、时延

2002年8月28日收稿

* 国家自然科学基金资助项目,编号:60072036

** 男 27岁 博士生

变化大三点。TCP的一个基本假设认为,所有的丢包都是因网络拥塞引起的,每检测到丢包时,都会启动相应的拥塞控制机制,减小发送窗口的大小,以降低TCP的发送速率。这个假设在有线网络中是合理的,但在无线网络中却不合适。无线链路上较高的误码率和移动主机的频繁切换,会导致大量的丢包,一旦发生丢包,TCP会启动不必要的拥塞控制机制,造成链路空闲,浪费发送机会,导致系统性能下降。

2 相关工作

目前,研究改善无线链路上的TCP性能的相关工作如图1所示。即移动主机(Mobile Host, MH)通过无线链路与基站相连,再通过基站接入Internet,与某一固定主机(Fixed Host, FH)通信,从MH到固定网络通常只有一跳的距离。

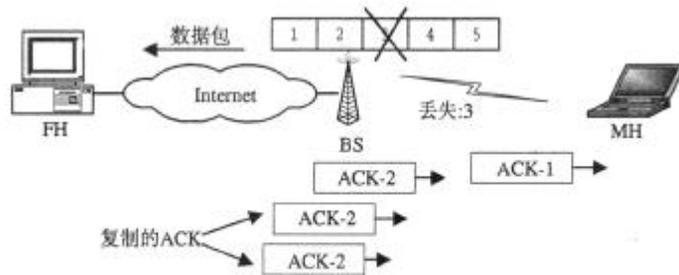


图1 MH通过无线链路接入Internet

2.1 链路层方案

为了克服无线链路高误码率和高丢包率的缺点,许多无线接入技术都在链路层采用了前向纠错(Forward Error Correction, FEC)和自动重传请求(Automatic Repeat request, ARQ)的技术来尽可能地保证链路层的可靠传输。链路层方案也不可能完全消除无线链路上的误码和丢包,数据通信的可靠性最终要靠传输层协议来保证。链路层和传输层的重传会相互作用,有可能使整体性能更加恶化^[4]。

2.2 连接分段方案

从发送端到接收端的传输层连接在基站处被分成无线链路和固定链路两段。文献[5]提出在无线链路上采用一种专用于无线链路的传输层协议(Selective Repeat Protocol, SRP),它运行在UDP之上,但是文献[5]的仿真结果表明,TCP的性能并没有得到明显的改善。文献[6, 7]提出Indirect TCP(I-TCP)方案,将无线链路和固定链路隔离开,在无线链路上仍采用TCP协议。仿真结果表明该方案也没得到明显的性能改善。连接分段方案的缺点是它破坏了传输层协议端到端的语义,基站需要为每一条连接维持大量的状态信息,处理所有的数据包,开销很大。基站处需要大量的缓存空间来存放数据包。

2.3 Snoop方案

Snoop方案文献[8]是一种利用了传输层知识的链路层协议。在基站处设置一个软件模块 - Snoop代理(Snoop Agent)。该代理监视每条TCP连接的每一个包 - TCP报文段和ACK,同时缓存所有未应答的TCP报文段。该代理通过收到多个重复的ACK,发生了本地超时而判断出某个报文段在无线链路上丢失,将缓存中的TCP报文段重传,扔掉由该报文段丢失引起重复的ACK,既能够将无线链路上丢失的报文段恢复,又防止了发端TCP收到多个重复的ACK而启动快速重传。Snoop方案适用于改善从固定主机到MH方向上的性能。

为了改善从MH到FH方向的TCP性能,文献[9]提出了一种显式丢失通知(Explicit Loss Notification, ELN)方案,它也是在基站处设置一个TCP代理,监视所有从MH发往FH的TCP报文段,记录下在无线链路丢失的报文段的序号,在返回MH的ACK中设置一些比特来标明这些报文段已经丢失,指示MH重传这些报文段。ELN方案需要更改MH中的TCP实现。

3 无线链路TCP性能的改进算法

为改善从MH到FH方向上的TCP性能,本文提出,在无线链路靠近固定网络的一侧,即基站(Base Station, BS)处,为每一条TCP连接设置一个TCP代理,类似于文献[8]中的Snoop代理。TCP代理监视所有从MH发往

FH的TCP报文段,记录收到报文段的序号,一旦出现漏洞,即可推断出报文段在无线链路上丢失,并确定其序号。同时,该TCP代理还监视所有由FH发往MH的ACK,一旦发现ACK应答的报文段为已丢失的报文段的前一个,则TCP代理立即转发该ACK,同时复制两份,立即发送给MH,迫使MH快速重传已丢失的报文段,防止了MH进入超时重传,从而改善了TCP的性能。对于后续的重复的ACK,TCP代理不再处理,直接转发给MH。

需要注意的是,虽然BS可以根据收到的报文段序号推断出丢失的报文段,但是却不能立即发送ACK给MH。即使在无线链路上没有丢失的报文段,也可能因为网络拥塞最终没有到达FH。且ACK是累积的,即后面的ACK包含了之前的ACK信息,如BS贸然发送ACK给MH,则有可能破坏TCP端到端的语义,此时被应答的TCP报文段可能还未被FH正确接收。

如图1所示,BS收到报文段1,2,4,5,判定报文段3在无线链路上丢失。当ACK-2到达时(表明FH已正确接收到报文段2,请求MH发送报文段3),BS复制两个ACK-2并发送给MH,迫使MH进入快速重传阶段。需指出的是,TCP中的ACK的序号是已接收到字节的下一个字节的序号,因为NS文献[10]中采用它所应答的报文段的序号来表示,在本文例子中以及在后面仿真结果中,也采用类似的表示方法。方案的益处有:

1) 通过复制两个ACK迫使MH立即进入快速重传,节省了MH端TCP等待重传的时间。通过监视TCP报文段,确定了已丢失的报文段,这时需要MH尽可能快的重传该报文段,防止不必要的超时重传。复制2个ACK,使MH端的TCP获得尽可能多的发送新数据机会。

2) 在应用中,大量存在小窗口问题^[8]。文献[11]通过对访问1996年亚特兰大夏季奥林匹克运动会官方网站的Web服务器165万条TCP连接进行统计分析,结论是:在所有由丢包引起的重传中,43.8%是快速重传,56.2%是至少一次超时引起的重传。有一半以上的丢包由超时重传来恢复,一个重要原因就是缺乏足够的ACK来触发快速重传。由于TCP的发送窗太小,导致中间链路上的TCP报文段的数目太少,不能够产生足够的ACK来触发快速重传或者是不能在超时之前触发快速重传。在无线应用中,无线链路的带宽通常很小,小窗口问题会更加严重。在本文的方案中,由于复制了ACK,所以能够有效的防止由于ACK不足引起的超时重传。

4 计算机仿真结果与分析

使用NS文献[10]软件对本文方案进行了计算机仿真,网络拓扑结构采用图1所示的结构,使用TCP Reno,每个TCP报文段的大小为1 000字节。

4.1 TCP报文段的轨迹图

本文设置固定链路的带宽为5 Mb/s,时延为300 ms。在仿真中,采用加载了错误模型的固定链路来模拟高丢包率的无线链路,这段链路的带宽也设为5 Mb/s,时延为20 ms;其中错误模型是均匀分布的丢包模型,丢包率可设。这样做是因为:

1) NS中的无线链路模型还不够成熟;
2) 仿真的目的是验证本文方案在有丢包的情况下的作用,不限于无线链路,可用于链路质量很差的有线链路。

因此,采用固定链路加丢包模型来模拟无线链路。

本文首先设TCP接收端广播窗口为3,丢包率为10%,分别得到TCP Reno和本文方案在前10 s的TCP报文段的轨迹图,如图2所示。在图2a中,报文段4和报文段10,11在无线链路上丢失,都引起了超时重传;在图2b中,基站很快检测出报文段4丢失,在接收到ACK-3时,复制了两个ACK-3,并发送给MH,迫使MH快速重传报文段4。需要指出的是,由于3个ACK-3在时间轴上相距太近,图中只看到一个圆圈,所以对这三个ACK做了标注。在图2b中,报文段10,11在无线链路上丢失,在时刻3.98 s,ACK-9到达BS,此时BS还没有报文段10,11的信息,它没有复制ACK-9,只有在时刻4 s,收到报文段12时,才判断出报文段10,11已经丢失,所以在时刻4.62 s,BS再次收到ACK-9,这时它复制并重传ACK-9,使得MH立即重传报文段10。在时刻5.26 s,BS收到ACK-10,它也复制并重传了3个ACK-10,但是MH收到三个重复的ACK后并没有立即重传报文段11,报文段11是在时刻6.47 s由于TCP超时重传的,这种情况是由于TCP Reno实现限制所造成的。

在TCP Reno中, 当一个窗口中有多个报文段丢失且距离很近, 第一个丢失的报文段可以通过快速重传恢复, 而其他的, 则一般需要通过超时重传才能恢复^[8], 因此, 报文段11通过超时重传得到恢复。

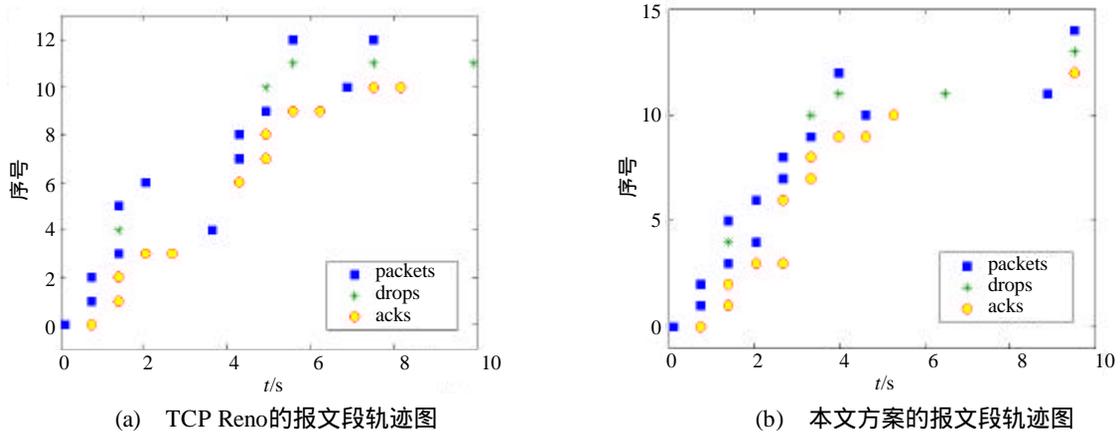


图2 报文段轨迹图(wnd=3)

TCP Newreno文献[12]正是为了解决这个问题而提出的。为了验证这一点, 本文在TCP Newreno下进行了仿真, 得到的前10 s的TCP轨迹如图3所示。从图3中可以看出, 报文段11得到了快速重传, 不幸的是, 在无线链路上再次丢失。需要指出的是, 虽然BS在5.90 s再次发送了3个ACK, 此时即使是TCP Newreno也无能为力, 不再快速重传, 必须等待超时了。

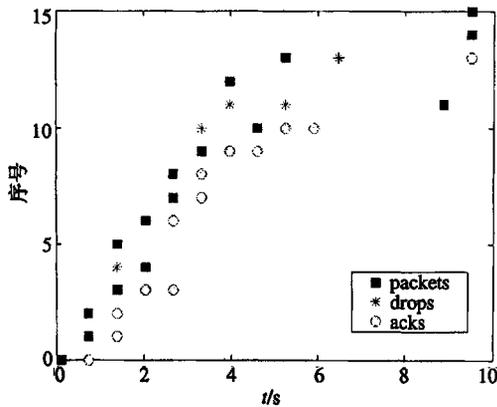


图3 本文方案在TCP Newreno下的TCP报文段轨迹图

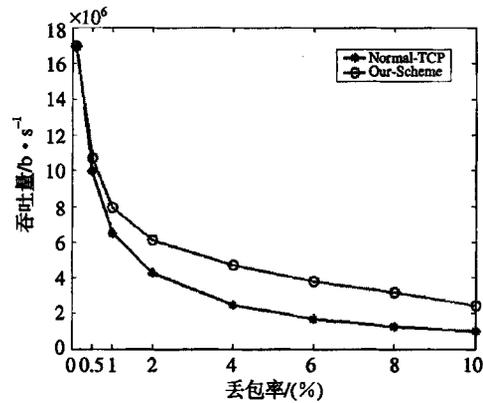


图4 LAN条件下的性能(wnd=32)

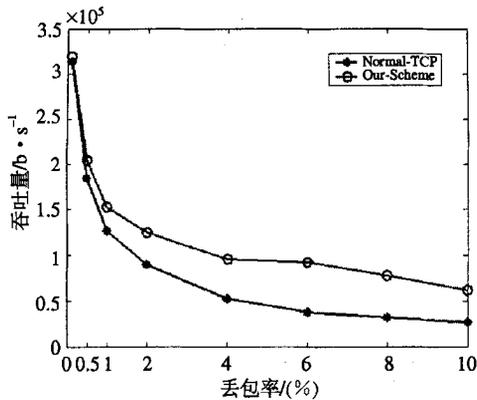


图5 WAN条件下的性能(wnd=32)

4.2 不同网络条件下的TCP吞吐量

考察在不同的网络条件下本文方案的性能。首先模拟局域网(LAN)条件下的性能。设固定链路的时延为50 ms, 无线链路时延为10 ms, 接收端广播窗口wnd为32个报文段。得到的本文方案和TCP Reno在不同丢包率条件下的性能图如图4所示。从图中可以看出, 当丢包率较低时(小于0.5%), 性能改善不明显, 是因为此时丢包数目很少, 造成TCP 连接中断的时间很短, 丢包虽然通过本文方案加以快速恢复, 但是对平均吞吐量的改善还是不明显; 当丢包率较高时, 性能的改变就很明显了。

其次模拟了广域网(WAN)条件下的性能。设固定链路的时延为300 ms,无线链路时延为20 ms。得到该方案和TCP Reno在不同丢包率条件下的性能如图5所示。

6 结束语

本文对无线链路的TCP性能加以研究,提出了一种改善从MH到FH方向上TCP性能的方案,并进行了计算机仿真。无线链路上的TCP性能是无线应用中的一个非常关键的问题,在这方面的研究还需要深入的进行。

参 考 文 献

- [1] Jacobson V. Congestion avoidance and control[J]. Computer Communication Review, 1988, 18(4): 314-329
- [2] Postel J. Transmission control protocol[S]. IETF RFC 793, 1981
- [3] Stevens W. TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms[S]. IETF RFC 2001, Jan. 1997
- [4] Hari Balakrishnan, Venkat N. Padmanabhan, Srinivasan Seshan, *et al.* A comparison of mechanisms for improving TCP performance over wireless links[J]. IEEE/ACM Transactions on Networking, 1997, 5(6).
- [5] Yavatkar R, Bhagwat N. Improving end-to-end performance of TCP over mobile internetworks[A]. Mobile 94 Workshop Mobile Computing Syst. Appl., 1994
- [6] Ajay Bakre, Badrinath B R. I-TCP: Indirect TCP for mobile hosts[A]. Proc. 15th Int. Conf. on Distributed Computing Syst. (ICDCS), 1995
- [7] Bakre A, Badrinath B R. Implementation and performance evaluation of indirect-TCP[J]. IEEE Trans. Comp., 46(3), 260-278, 1997
- [8] Balakrishnan H. Challenges to reliable data transport over heterogeneous wireless networks[D]. Ph.D. thesis, 1998
- [9] Balakrishnan H, Katz R H. Explicit loss notification and wireless web performance[A]. Proc. Globecom98 (Internet Mini Conference), 1998
- [10] Ns2, Network Simulator, Version 2, home page(J/OL). <http://www.isi.edu/nsnam/ns>
- [11] Balakrishnan H, Padmanabhan V N, Seshan S, *et al.* TCP behavior of a busy web server: analysis and improvements[A]. In Proc. IEEE INFOCOM, 1998
- [12] Hoe J C. Improving the start-up behavior of a congestion control scheme for TCP[A]. Proc. ACM SIGCOMM' 96, 1996

编 辑 孙晓丹