Journal of Zhejiang University SCIENCE A
ISSN 1009-3095
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn

Modeling TCP performance with proxy and ARQ

JIN Yan-liang (金彦亮), YANG Yu-hang (杨宇航)

(*Modern Communication Institute, Shanghai Jiao Tong University, Shanghai 200030, China*)

E-mail: jinyanliang@staff.shu.edu.cn; yyang@info.sh.cn

Received Sept. 10, 2004; revision accepted Nov. 15, 2004

Abstract: TCP/IP is a next generation key technology in the wireless communication network where the different characteristics of wireless and wired links result in performance degradation. We can use the proxy and automatic repeat request (ARQ) schemes to deal with this problem. In this work, we investigate the TCP performance over proxy and ARQ in the wireless network. Our analysis results showed that using the proxy can result in lower transfer latency and higher throughput and that ARQ can decrease the loss rate of wireless link and improve the performance with little additional latency. The analytical results were validated against simulations using the NS-2 with some more realistic parameters.

Key words: TCP performance, Wireless communication, Proxy, ARQ

doi:10.1631/jzus.2006.A0164 **Document code:** A **CLC number:** TN919

INTRODUCTION

The convenience of wireless communications has led to increasing use of wireless networks for civilian and critical applications. With the advent of Internet technology and the increasing popularity of wireless data devices, the wireless industry is evolving its core networks toward IP-based networks. It will be necessary to integrate the wireless networks and the existing wired networks into the Internet.

TCP has been the predominant transport protocol used in the wired Internet to deliver data reliably and will be a key technology in the future Internet. It is well-known that TCP is not efficient in wireless networks due to their distinct features, e.g., the dynamics of link capacity, error control mechanisms, etc. These very different characteristics of wireless and wired links result in performance degradation. In order to deal with the heterogeneous environment problem and improve the performance, two schemes can be used. One is a TCP connection splitting and spoofing proxy that pre-acknowledges the sender on behalf of the receiver and forwards packets to the receiver on behalf of the sender. Such a scheme is usually called split TCP, TCP spoofing or Indirect TCP (I-TCP) (Bakre and Badrinath, 1995; 1997), which divides a single connection into wired and wireless parts and separates congestion losses from link failure losses. The other is error control scheme in the link layer. Common error control methods used in wireless communication include forward error correction (FEC), automatic repeat request (ARQ) and their hybrids (Cain and McGregor, 1997; Fantacci, 1996).

Researches on the analytical characterization of TCP were recently carried out (Padhye *et al.*, 1998; 1999) focused mainly on TCP throughput as a function of loss rate in wired networks. Liu and Ehsan (2004) developed a simple mathematical model to investigate the TCP performance over proxy, with analysis results implying that it is important to minimize the asymmetry between these parts of a connection (segregated by proxy), especially in a heterogeneous environment. Balakrishnan *et al.* (1997) evaluated the throughput performance of a single wireless TCP connection experimentally, with results showing that link layer retransmission hides bit error on a wireless link to higher layer protocols is effective. Canton and Chahed (2001) considered the end-to-end transmission over ARQ in UMTS environment with results showing that the end-to-end performance of TCP in a mobile, wireless setting is enhanced by the presence of ARQ.

In this paper, we propose an analytical model to determine the TCP performance in a wireless network environment where a proxy is in the middle of the wired and wireless links and the ARQ scheme is active at the wireless link of the end-to-end path. Our detailed analysis is mainly directed to the lossy scenario and we use some more relatively realistic parameters to evaluate the TCP performance. Our results showed that both proxy and ARQ could decrease the adverse effect of the wireless link and improve the system performance.

The remainder of this paper is organized as follows. In Section II, we present our network model upon which our analysis is based. In Sections III and IV, we analyze the latency in file transfer with and without using the proxy and ARQ under the lossless and lossy scenario, respectively. Section V discusses the latency and throughput of end-to-end and split TCP connection in a wireless network based on some more realistic parameters. Finally we give our conclusions and discuss in Section VI.

A MODEL FOR TCP CONGESTION CONTROL OVER PROXY AND ARQ

Network scenario

The reference network considered in this paper is depicted in Fig.1. A number of TCP senders (receivers) placed on mobile terminals are connected to some TCP receivers (senders) placed within the wired section of the Internet. As for wireless interface, we assume a centralized wireless access handled by a Base Station (BS). Mobile terminals communicate with wired terminals through the BS connected to the wired network with a designated link.

Network model

We propose a two-link model with one end node

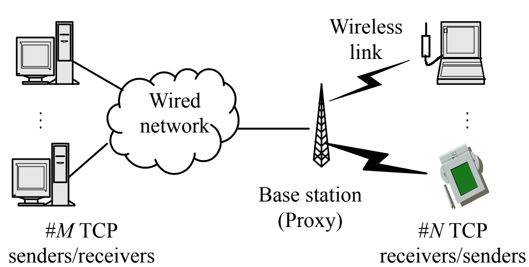

Fig.1 Network scenario

on each side and a proxy in the middle, as shown in Fig.2. The proxy is placed between a wired part and a wireless part, and the sender/receiver is located at the end of the wired/wireless link. In fact, in Fig.2a the proxy functions as a normal base station that forwards packets from the sender to the receiver and vice versa. Each of the two links may contain parts that are abstracted into a single link with a single round trip time (RTT) parameter and a single loss rate parameter. Two error control schemes are introduced independently into different layers: TCP at the transport layer and ARQ at the link layer. File transfer is our main application of interest and without loss of generality is considered to be from the wired terminal (TCP sender) to the wireless terminal (TCP sink). We consider a classical Go-back-N ARQ as described in (Bertsekas and Gallager, 1992), with ARQ source at the wireless terminal and ARQ receiver at the proxy (base station). We use silent ARQ scheme that does not disturb TCP and only introduces latency. The ARQ sender segments the TCP segment into constant-size ARQ frames and sends them in an ordered fashion and the receiver does not accept out-of-order frames. We employ only the ARQ, because due to the limitation of error correcting capability of FEC, ARQ is necessary to provide error free link even when we employ FEC.

Wireless channel error model

The wireless channel is modeled as Gilbert-Elliot channel (Zorzi *et al.*, 1998), widely used in literature to characterize wireless transmission medium.

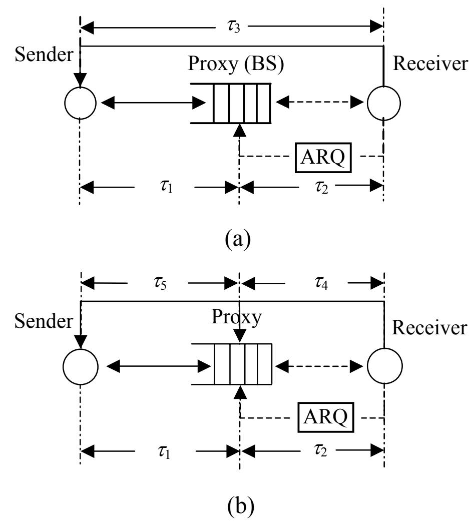

Fig.2 Network model. (a) End-to-end connection; (b) Split connection

The transmission medium is modeled as a two-state (Good and Bad) Markov chain. The transition probability matrix $\boldsymbol{M}_{\mathrm{p}}$ of the model can be given by:

$$\boldsymbol{M}_{\mathrm{p}} = \begin{bmatrix} c & q \\ r & s \end{bmatrix}, \tag{1}$$

where c is the probability of correlated success, s probability of correlated error, q probability of error after success and r probability of success after error. r and q take larger values for fast fading than for slow fading channels. When bit errors are independent, $c=r=1-e_{\mathrm{fer}}$ and $q=s=e_{\mathrm{fer}}$, where e_{fer} is the frame error rate.

Assumptions and parameters

For the convenience of tracing our analytical model, we list below some mathematical symbols that will be used in later sections: *MSS*, maximum segment size of TCP packet; M, the amount of *MSS* that a file contains; D, packet length; μ_1, μ_{p}, μ_2, the time that sender, proxy and receiver takes to transmit a packet; ω_0, $\omega_{\text{ss-thresh}}$, $\omega_{\max}$, initial window size, slow stat thresh and maximum window size of the sender; ω_0', $\omega_{\text{ss-thresh}}'$, $\omega_{\max}'$, initial window size, slow stat thresh and maximum window size of the proxy; t_{p}, the pre-packet processing delay at the proxy; $\tilde{\mu}_{\mathrm{p}}$, the time that the proxy takes to transmit a packet when ARQ is used; τ_1, τ_2, one-way propagation delay on the sender-proxy and proxy-receiver link; τ_3, the transmission latency of the corresponding ACK for the released segment; τ_4, τ_5, the latency of ACK on the receiver-proxy and proxy-sender link; τ_{ARQ}, the total ARQ frame processing delay.

We assume that the TCP sender is only constrained by the congestion control window and not the advertised receive window size. When the proxy is used, we have two serial connections that are not independent but coupled by data, because the second connection cannot send any data packets it has not received from the first connection. For simplicity, we will assume that the second connection is never constrained by the first connection in the following discussion.

DELAY ANALYSIS OVER LOSSLESS LINKS

Typical TCP behavior exhibits obvious cyclical evolution starting with a slow-start phase followed by congestion avoidance phase until the maximum window size is reached. Since all new TCP versions use delayed ACK, which means that 1 ACK is sent back for roughly every b packets. So the exponential growth rate of the congestion control window is $r=1+1/b$. If the file is big enough and the slow-start thresh is reached during the $(S+1)$th window, we obtain

$$\omega_0 r^{S-1} < \omega_{\text{ss-thresh}} \le \omega_0 r^S. \tag{2}$$

Similarly, if the file is big enough and the maximum congestion window size is achieved during the (M_x+1)th window, we have

$$\omega_{\text{ss-thresh}} + \frac{M_x - S - 1}{b} < \omega_{\max} \le \omega_{\text{ss-thresh}} + \frac{M_x - S}{b}. \tag{3}$$

All subsequent windows have the same window size $\omega_{\max}$. The number of windows needed to transfer an M-segments file is given as follows:

$$K = \begin{cases} \min\left\{k: \sum_{i=1}^{k} \omega_0 r^{i-1} \ge M\right\}, & k \le S \\ \min\left\{k: \sum_{i=1}^{S} \omega_0 r^{i-1} + \sum_{i=S+1}^{k}\left(\omega_{\text{ss-thresh}} + \frac{i-S-1}{b}\right) \ge M\right\}, & S < k \le M_x \\ \min\left\{k: \sum_{i=1}^{S} \omega_0 r^{i-1} + \sum_{i=S+1}^{M_x}\left(\omega_{\text{ss-thresh}} + \frac{i-S-1}{b}\right) + \sum_{i=M_x+1}^{k} \omega_{\max} \ge M\right\}, & M_x < k \end{cases} \tag{4}$$

So, the time that it takes to transmit the kth window at the sender can be given by

$$t_k(\mu_1) = \begin{cases} \omega_0 r^{k-1} \mu_1, & k \le S \\ \left(\omega_{\text{ss-thresh}} + \frac{k-S-1}{b}\right)\mu_1, & S < k \le M_x \\ \omega_{\max}\mu_1, & M_x < k \end{cases} \tag{5}$$

Delay of an end-to-end connection

When the links are lossless, the ARQ scheme is not triggered. If $\mu_1 \ge \mu_{\mathrm{p}}$, which indicates that the proxy can transmit as fast as the sender, the round-trip time of end-to-end connection is ($\tau_1+\tau_2+\tau_3$). The time it takes for the first ACK to arrive after the first packet was sent is $b\mu_1+\mu_p+\tau_1+\tau_2+\tau_3$. The total time it takes to transfer an M-segments file is then:

$$T_e(M) = M\mu_1 + \tau_1 + \tau_2 + \mu_p + \sum_{k=1}^{K-1}[b\mu_1 + \mu_p + \tau_1 + \tau_2 + \tau_3 - t_k(\mu_1)]^+,$$

where $[\alpha]^+=\alpha$ if $\alpha>0$ and $=0$ otherwise; and the receiver returns an ACK every b packets. K can be calculated by substituting the sender's initial window size, slow-start thresh and maximum window size into Eq.(4). If $\mu_1<\mu_p$, packets experience additional queuing delay at the proxy and the latency should be computed from the receiver's side. The receiver gets packets of the same window continuously at rate $1/\mu_p$ and the idle time of the receiver is $[\mu_1+\mu_p+\tau_1+\tau_2+\tau_3+(b-1)\mu_p-t_k(\mu_p)]^+$, where $t_k(\cdot)$ can be given by Eq.(5). The total latency is then:

$$T_e(M) = M\mu_p + \tau_1 + \tau_2 + \mu_1 + \sum_{k=1}^{K-1}[b\mu_p + \mu_1 + \tau_1 + \tau_2 + \tau_3 - t_k(\mu_p)]^+.$$

We can get the latency of transmitting a file of M-segments using the end-to-end connection and get the same result as that in (Liu and Ehsan, 2004), that is

$$T_e(M) = \min(\mu_1,\mu_p) + \tau_1 + \tau_2 + M\max(\mu_1,\mu_p) + \sum_{k=1}^{K-1}[R_e - t_k(\max(\mu_1,\mu_p))]^+, \quad (6)$$

where $[\alpha]^+=\alpha$ if $\alpha>0$ and $=0$ otherwise. Substituting the sender's initial window size, slow-start thresh and maximum window size into Eq.(4), we can obtain the K value. $R_e=\min(\mu_1, \mu_p)+b\max(\mu_1, \mu_p)+\tau_1+\tau_2+\tau_3$ and $t_k(\cdot)$ is given by Eq.(5).

Delay of a split connection

When the proxy is used, we have two serial connections and the window of the second connection evolves not only according to the window dynamics of TCP, but also according to the availability of packets from the first connection. We only study the case where the sender is fast enough so that the proxy is never constrained by unavailability of data. By following the same analysis as that in (Liu and Ehsan, 2004), the total transfer latency of an M-segments file for the proxy case can be given by

$$T_p(M) = \mu_1 + \tau_1 + t_p + \sum_{k=1}^{K'-1}[R_2 - t_k(\mu_p)]^+ + \tau_2, \quad (7)$$

where $[\alpha]^+=\alpha$ if $\alpha>0$ and $=0$ otherwise. K' can be calculated by substituting the proxy's initial window size, slow-start thresh and maximum window size into Eq.(4). $R_2=b\mu_p+\tau_2+\tau_4$ and $t_k(\cdot)$ can be given by Eq.(5).

ANALYSIS OF DELAY OVER LOSSY LINKS

In wireless medium, error is significant and more responsible for service degradation than loss. When losses occur, ARQ scheme is employed to detect and correct the link-level error and the detailed analysis will become more complicated.

The throughput of a TCP connection is well-known to depend upon packet loss probability, round-trip time (RTT) and timeout (Padhye *et al.*, 1998). The expression in (Padhye *et al.*, 1998) for the steady-state TCP throughput, in a wired context only, as a function of loss probability p is as follows:

$$[Th(RTT,p)]^{-} = RTT\sqrt{\frac{2bp}{3}} + T_0\min\left(1, 3\sqrt{\frac{3bp}{8}}\right)p(1+32p^2), \quad (8)$$

where RTT is the TCP round trip time, T_0 is the TCP timeout and b is the number of TCP segments sent back-to-back and for which only one cumulative ACK is generated. It also is equally effective when applied to short TCP connections if combined with slow-start phase analysis (Liu and Ehsan, 2004).

In what follows we will study the case where the sender-proxy link is lossless, then study the case where both links are lossy.

Sender-proxy link is lossless

1. Delay of an end-to-end connection

We assume that the sender-proxy link is lossless. Similar to (Liu and Ehsan, 2004), let us denote the loss probability on the proxy-receiver link as p_2, and the throughput as $Th(RTT, p)$. The latency it takes to transfer this M-segments file using end-to-end connection is

$$\begin{aligned} T_e &= \sum_{n=0}^{M} p(n)\left(T_e(n) + \frac{M-n}{Th(RTT, p_2)}\right) \\ &= \sum_{n=1}^{M} p(n)T_e(n) + \frac{M - m_{\text{loss}}}{Th(\tau_1+\tau_2+\tau_3, p_2)}, \end{aligned} \quad (9)$$

where $p(n)=(1-p)^n p$ if $n<M$ and $p(n)=(1-p)^M$ if $n=M$ are the probabilities that n packets are successfully sent before the first loss occurs. $m_{\text{loss}}=(1-(1-p_2)^M)\times(1-p_2)/p_2$ is the expected number of packets sent before the first loss occurs and $T_e(\cdot)$ can be given by Eq.(6).

We assume that ARQ can recover all the corrupted frames and there is no congestion loss on the wireless link, so the wireless link becomes lossless. The ARQ transmissions and retransmissions introduce additional latency in processing frames and sending them through the wireless channel. So, we have

$$\tilde{\mu}_p = \mu_p + E(\tau_{\text{ARQ}}) = \mu_p + nbD_{\text{ARQ}} + ND_{\text{ARQ}}\frac{q(nb-1)}{1-s^N}, \quad (10)$$

where D_{ARQ} is the time for processing one ARQ frame, which is related to the RTT between the proxy and the receiver, q and s are defined in Eq.(1), one TCP segment is divided into n ARQ frames with n being an integer.

We use $\tilde{\mu}_p$ to replace μ_p in Eq.(6) and get the latency as follows:

$$\begin{aligned} T_e^1(M) &= \min(\mu_1, \tilde{\mu}_p) + \tau_1 + \tau_2 + M\max(\mu_1, \tilde{\mu}_p) \\ &\quad + \sum_{k=1}^{K-1}[R_e - t_k(\max(\mu_1, \tilde{\mu}_p))]^+, \end{aligned} \quad (11)$$

where $[\alpha]^+=\alpha$ if $\alpha>0$ and $=0$ otherwise. Substituting the sender's initial window size, slow-start thresh and maximum window size into Eq.(4), we can obtain the K value. $R_e=\min(\mu_1, \tilde{\mu}_p)+b\max(\mu_1, \tilde{\mu}_p)+\tau_1+\tau_2+\tau_3$ and $t_k(\cdot)$ is given by Eq.(5).

In fact, the wireless links are not fully shielded from random errors caused by the non-ideality of the wireless channel even when we employ ARQ scheme. Denoting the loss rate on the proxy-receiver link with ARQ by p_2' ($p_2' < p_2$), and throughput by $Th(RTT', p_2')$, the transfer latency of a file of M-segments size can be easily computed as

$$\begin{aligned} T_e^2 &= \sum_{n=0}^{M} p'(n)\left(T_e(n) + \frac{M-n}{Th(RTT', p_2')}\right) \\ &= \sum_{n=1}^{M} p'(n)T_e(n) + \frac{M - m'_{\text{loss}}}{Th(\tau_1+\tau_2+\tau_3+E(\tau_{\text{ARQ}}), p_2')}. \end{aligned} \quad (12)$$

2. Delay of a split connection

When the proxy is used and the proxy-receiver connection is not constrained by the sender-proxy connection, the transfer latency of a file of M-segments size can be easily computed as

$$T_p = \sum_{n=1}^{M} p(n)T_p(n) + \frac{M - m_{\text{loss}}}{Th(\tau_2+\tau_4, p_2)}. \quad (13)$$

When ARQ is used to recover all the corrupted frames and the wireless link is lossless, we can calculate the transfer latency as follows:

$$T_p^1(M) = \mu_1 + \tau_1 + t_p + \sum_{k=1}^{K'-1}[R_2 - t_k(\tilde{\mu}_p)]^+ + \tau_2, \quad (14)$$

where $\tilde{\mu}_p$ is give by Eq.(10) and $R_2=b\tilde{\mu}_p+\tau_2+\tau_4$.

If ARQ cannot recover all the error and the loss rate on the proxy-receiver link becomes p_2' ($p_2' < p_2$), we can obtain

$$T_p^2 = \sum_{n=1}^{M} p'(n)T_p(n) + \frac{M - m'_{\text{loss}}}{Th(\tau_2+\tau_4+E(\tau_{\text{ARQ}}), p_2')}. \quad (15)$$

Both links are lossy

1. Delay of an end-to-end connection

Assuming the loss rates on the sender-proxy link and the proxy-receiver link are p_1 and p_2 and independent, the overall loss rate experienced by an end-to-end connection is $p=p_1+p_2-p_1p_2$. If the file is big enough, the end-to-end connection throughput of this case is $Th(\tau_1+\tau_2+\tau_3,p)$ and the latency of an M-segment file can be given by Eq.(9). If the wireless link becomes lossless when ARQ is used, the overall loss late becomes p_1, the throughput of this case becomes $Th(\tau_1+\tau_2+\tau_3+E(\tau_{\mathrm{ARQ}}),p_1)$ and the total transfer latency can be calculated by Eq.(12). If the wireless link is still lossy when ARQ is used and the loss rate on the proxy-receiver link is p_2', the overall loss rate will become $p'=p_1+p_2'-p_1p_2'$. Then the latency is given by Eq.(9) and the end-to-end connection throughput can be calculated by $Th(\tau_1+\tau_2+\tau_3+E(\tau_{\mathrm{ARQ}}),p')$.

2. Delay of a split connection

When the proxy is used, the proxy-receiver connection is not constrained by the sender-proxy connection and the file is big enough, the transfer latency of a file of M-segments size can be computed as Eq.(13), where the loss rate is $p=p_1+p_2-p_1p_2$ and the throughput is $\min(Th(\tau_1+\tau_5,p_1), Th(\tau_2+\tau_4,p_2))$. If the ARQ does correct all errors, the overall loss rate would become p_1 and the total transfer latency can be calculated by Eq.(15), where the throughput is $Th(\tau_1+\tau_5+E(\tau_{\mathrm{ARQ}}),p_1)$. If ARQ does not correct all the corrupted frames and the loss rate on the proxy-receiver link is p_2', and the overall loss rate becomes $p'=p_1+p_2'-p_1p_2'$. Then the latency is given by Eq.(13) and the throughput can be calculated by $\min(Th(\tau_1+\tau_5,p_1), Th(\tau_2+\tau_4+E(\tau_{\mathrm{ARQ}}), p_2'))$.

SIMULATION AND DISCUSSION

We use NS-2 (McCanne and Floyd, 1997) to obtain the simulation results. One NS-2 limitation that most affected our work is that every node must be configured as an agent or a sink, but not both. We need such a node as the proxy. However, NS-2 does not permit this and some changes are required. We use the buffer copy to realize it. We simulate a topology of a network with a wired portion including a 10 Mbps link between a source node and a base station. The propagation time over the wired link is initially assumed to be 45 ms. Later, the propagation time is varied from 0 to 250 ms to represent a variety of wired network environments ranging from campus to intercontinental connections. The wireless portion of the network is a very short 2 Mbps wireless link with a propagation time of 0.01 ms. The wireless link is assumed to connect the base station to a mobile terminal. The ARQ parameters are n=10, b=2, N=10 and D_{ARQ}=0.001 ms.

Congestion control window

We compared the sender congestion window (SCW) evolution of an end-to-end connection and a split connection with/without the ARQ scheme, as shown in Fig.3. When the proxy is used, the sender only sees the loss rate p_1 on the sender-proxy link, which is smaller than the loss rate of the end-to-end connection, $p=p_1+p_2-p_1p_2$. Thus, the steady-state window is larger in the split connection case, as shown in Fig.3a. When using the ARQ, we assume that the loss rate in the wireless link becomes p_2' ($p_2'\approx p_1$), so the sender window of the end-to-end case increases, thus justifying Fig.3b. Fig.3 also shows that ARQ results in shorter congestion control transient phase since it enhances the performance of the wireless link.

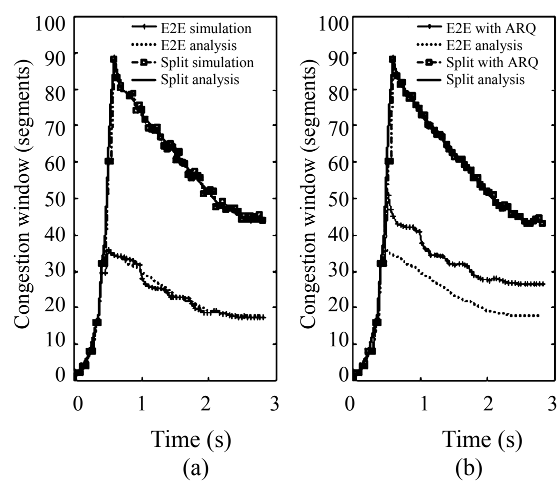

Fig.3 Sender congestion window of end-to-end and split connection for the case where τ_1=45 ms, τ_2=0.01 ms, p_1=0.01, p_2=0.05. (a) SCW without ARQ; (b) SCW with ARQ

Error rate of the wireless link

The simulation verified that, when the packet loss experienced by the TCP connection in the wireless network is one order of magnitude lower than the packet loss experienced by the wired network, TCP performance is essentially not affected by the wireless impairment, and is the same as the result in related work. Fig.4 shows that the presence of ARQ scheme considerably improves the TCP throughput with a little additional latency under a lower packet loss probability. The performance increases less when the proxy is used, but the ARQ can result in higher throughput and slower degradation. Though Go-back-*N* ARQ does not accept out of order packets which cannot cause a triple-duplicate ACK event, a higher packet loss probability can make the retransmission in wireless network and timeout occur frequently and Eq.(9) cannot be used to obtain the latency accurately, so that this results in the decreases of the throughput and the increases of the latency and the large differences between simulation and analysis lines.

Propagation delay of the wired link

We ran simulation with the wired link propagation time varying from 0 to 250 ms. The results in Fig.5 show that the larger propagation delay decreases the system performance and increases the system latency. The proxy and ARQ schemes can provide the performance gain, but the gain is limited when the two links are significantly different. At the same time, we can also see that the simulation and analysis lines overlap almost completely and our analysis is very accurate when one-way propagation delay of the wired link changes.

Correlated and random errors

We compared the performance of a random wireless channel to the performance of a correlated-corruption one, in Case (1) one correlated-error case of random errors, as shown in Fig.6. The correlated-error channel can achieve higher performance because the Go-back-*N* ARQ mechanism will retransmit *N* frames when every error occurs. Thus, the link is used and in Case (2) one random-error link is used. We can see that the throughput is lower in the benefit of ARQ is more significant when the correlation degree is higher. With the increases of the packet

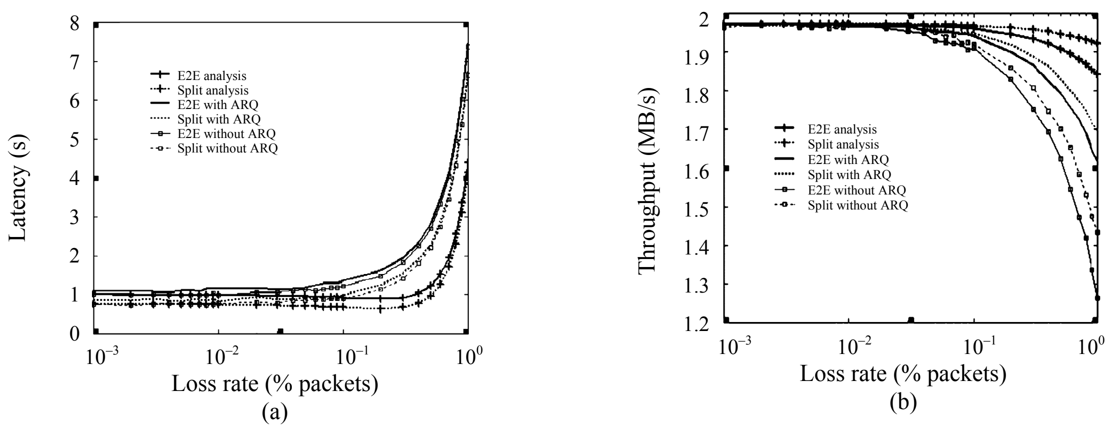

Fig.4 Latency (a) and throughput (b) vs error rate of the wireless link for the case where τ_1=45 ms, τ_2=0.01 ms, p_1=0.01

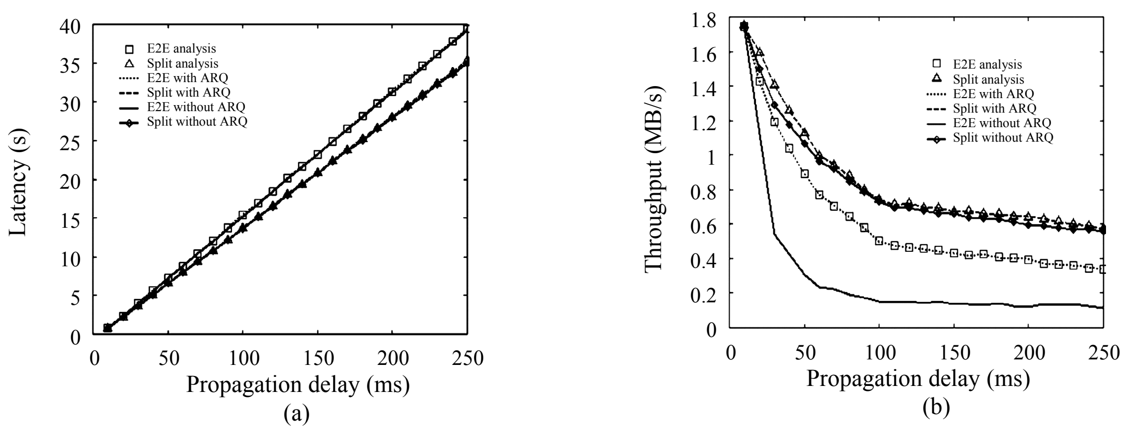

Fig.5 Latency (a) and throughput (b) vs one-way propagation delay for the case where τ_2=0.01 ms, p_1=0.01, p_2=0.01

loss probability, timeout occurs and the throughput decreases and the latency increases rapidly, as shown in Fig.4.

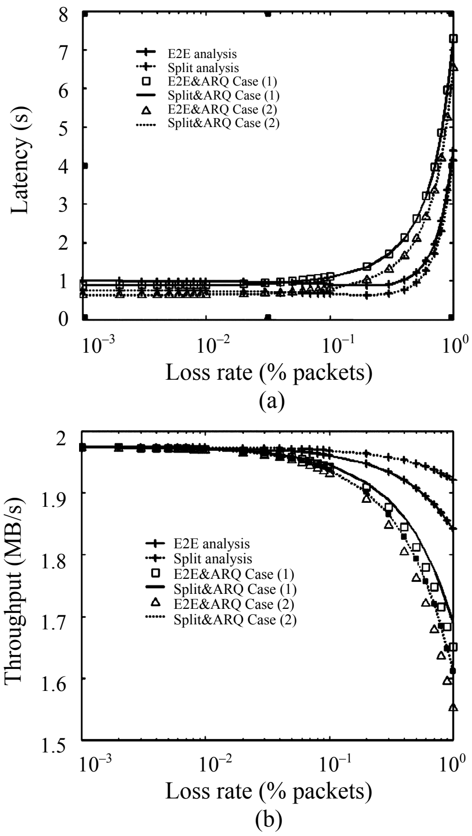

Fig.6 Latency (a) and throughput (b) vs different error patterns

CONCLUSION

In this paper, we proposed an analytical model for the TCP performance on a heterogeneous end-to-end path, composed of both wireless and wired links. The very different characteristics of the two kinds of links degrade the TCP performance dramatically. The proxy and ARQ schemes are used to decrease the asymmetry and improve the performance. Numerical results obtained from theoretical analysis and simulation showed that (1) using the proxy can result in lower transfer latency and higher throughput, but the performance gap between using a proxy and using an E2E connection becomes smaller as the level of link asymmetry increases; (2) ARQ can decrease the loss rate of wireless link and improve the performance with a little additional latency; (3) Using the proxy and the ARQ can decrease the asymmetry of the wired/wireless links and increase the overall performance. The presence of ARQ can enhance the end-to-end and split TCP performance and get the higher throughput when bit errors are heavily correlated each other.

We should note that further analysis could be conducted to deal with the latency of the high packet loss probability using different types of ARQ schemes, such as SR-ARQ. The hybrid error correction of ARQ and FEC also need to be focused on in the wired/wireless hybrid environment. Moreover, the non-persistent (short-lived) TCP connection problem and the different network environment are also our interesting tasks in future work.

References

Bakre, A., Badrinath, B.R., 1995. I-TCP: Indirect TCP for Mobile Hosts. Proceedings of the 15th International Conference on Distributed Computing Systems (ICDCS'95), IEEE Computer Society, Washington DC, USA, p.136-143.

Bakre, A., Badrinath, B.R., 1997. Implementation and performance evaluation of indirect TCP. *IEEE Trans. Computers*, **46**(3):260-278. [doi:10.1109/12.580423]

Bertsekas, D., Gallager, R., 1992. Data Networks (Second Edition). Prentice Hall, Inc., NJ, USA.

Balakrishnan, H., Padmanabhan, V.N., Seshan, S., Katz, R.H., 1997. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Trans. Networking*, **5**(6):756-769. [doi:10.1109/90.650137]

Cain, J.B., McGregor, D.N., 1997. A recommended error control architecture for ATM networks with wireless links. *IEEE Journal on Selected Areas in Communications*, **15**(1):16-28. [doi:10.1109/49.553675]

Canton, A.F., Chahed, T., 2001. End-to-end Reliability in UMTS: TCP over ARQ. IEEE Globecom'2001, San Antonio, p.3473-3477.

Fantacci, R., 1996. Queuing analysis of the selective repeat automatic repeat request protocol wireless packet networks. *IEEE Trans. Vehicular Technology*, **45**(2): 258-264. [doi:10.1109/25.492849]

Liu, M., Ehsan, N., 2004. Modeling TCP performance with proxies. *Journal of Computer Communications (Special Issue on Protocol Engineering for Wired and Wireless Networks)*, **27**(10):961-975.

McCanne, S., Floyd, S., 1997. ns-LBL Network Simulator. http://www.isi.edu/nsnam/ns/.

Padhye, J., Firoiu, V., Towsley, D., Kurose, J., 1998. Modeling TCP throughput: a simple model and its empirical validation. *ACM SIGCOMM Computer Communication Review*, **28**(4):303-314. [doi:10.1145/285243.285291]

Padhye, J., Firoiu, V., Towsley, D., 1999. A Stochastic Model of TCP-reno Congestion Avoidance and Control. CMPSCI Technical Report 99-02, Department of Computer Science, University of Massachusetts, Amherst.

Zorzi, M., Rao, R.R., Milstein, L.B., 1998. Error statistics in data transmission over fading channels. *IEEE Trans. Commun.*, **46**(11):1468-1477. [doi:10.1109/26.729391]