Interactions of TCP and Radio Link ARQ Protocol*

Yong Bai, Andy T. Ogielski†, and Gang Wu‡

WINLAB, Rutgers University
73 Brett Road, Piscataway, NJ 08854-8060
e-mail: [yongbai, ato, g-wu]@winlab.rutgers.edu

Abstract - Two distinct error and flow control schemes may be simultaneously involved in wireless Internet networks. One is an end-to-end transport protocol TCP which provides reliable packet delivery with flow and congestion control, and the other is a radio link ARQ protocol which provides radio link layer frame error recovery. In this paper, we investigate interactions between these two separate error control mechanisms by simulating TCP Reno over IS-707, a radio link ARQ protocol standard for spread spectrum digital cellular systems. We show the effects of TCP sliding window size to the overall system performance under a variety of radio link scenarios, and present the benificial effects of longer persistence in radio link frame error recovery for best-effort bulk data applications.

I. Introduction

A number of design incompatibilities between the Internet and wireless protocols have begun to emerge as the Internet connectivity reaches out to the mobile users of cellular systems. TCP has been designed under the assumption that packet losses are caused almost exclusively by network congestion, so TCP packet losses invoke congestion avoidance mechanisms [1, 2] incorporating rate reduction and multiplicative increase of the retransmission timeout. In a high and correlated radio link involved TCP connection, misinterpretation of packet losses over radio links as congestion losses leads to significant throughput degradation. Lower-level wireless protocols provide a number of error control methods, such as FEC (forward error correction), ARQ (automatic repeat request), and hybrid FEC/ARQ, to improve communication reliability. Link layer ARQ error control can reduce frame losses and improve the overall performance as specified in Radio Link Protocol (RLP) IS–707 [3] (an extended version of IS–99 [4]). However, if RLP and TCP operate independently of one another, their interactions need to be investgated in order to make them to cooperate more efficiently.

Several aspects of interactions between TCP and IS-707 RLP ARQ protocol have been studied in previous works[5] [6][7]. It was shown in [5] that with correlated losses slow frame recovery (without idle frames) as well as unrecovered errors cause successive TCP timeouts that trigger exponential TCP retransmission timer back-offs and yield long timeout value. Consequently, TCP waits a long idle timeout interval to discover that one packet has been lost and retransmite the awaited TCP packet. It was seen that in the high FER regime, the long idle waiting time on the transmission link is one reason for poor overall system performance. TCP can help RLP error recovery by lowering the round-trip timeout (RTO) upper bound and setting a smaller timer back–off factor, to the extent that it does not lead to congestion collapse.

In [6], we showed the improvement of TCP goodput by a fast RLP frame error recovery scheme based on using the IS-707 idle frames. We demonstrated how the use of idle frames in the high correlated frame loss regime can improve the useful throughput (goodput) of TCP data transfers. To achieve the required performance, the appropriate number of idle frames under various channel conditions can be obtained from the simulation results.

In this paper, we further investigate the performance and interactions caused by TCP sliding window size and persistence of RLP error recovery.

II. Effects of TCP Sliding Window Size

TCP uses a sliding window protocol for flow control and allows the window size vary over time. The sender can transmit up to the minimum of the congestion window (*cwnd*) and the advertised window (*awnd*). The congestion window is flow control imposed by the sender, while the advertised window is flow control imposed by the receiver. The former is based on the sender's assessment of

*Research of Y. Bai and A. T. Ogielski was supported by DARPA Grant N66001-96-C-8530 and by NSF Grant NCR-9527163.

†Also at the DIMACS Center, Rutgers University.

‡Also at Communications Research Laboratory, MPT, Japan.

perceived network congestion; the latter is related to the amount of available buffer space at the receiver for this connection[8].

For an error-free connection, in order to achieve maximum throughtput, the receiver's advertised window must be at least the capacity of the connection pipe, i.e., bandwidth-delay product. If advertised window is larger than the bandwidth-delay product, the throughtput remains at the maximum value for an error-free connection. However, when a slow lossy radio link is involved, if the advertised window is too large, there are two detrimental effects to the overall system performance. First, the TCP packets build up at the Internet Access Point (IAP), occupying large amount of buffer space and perhaps resulting in congestions. Secondly, when TCP packet is lost in the radio link, the retransmitted TCP packet is appended at the tail of the IAP buffer, thus the awaited TCP packet takes a long time to be delivered to the receiver delayed by the IAP FIFO queue, resulting in TCP timeouts and performance degradation.

On the other hand, if the advertised window is too small, there are no sufficient TCP packets available for delevery. This deficit degrades the overall performance as well, so a suitable advertised window size should be chosen to achieve desirable performance. The effects of TCP advertised wnidow size to the overall TCP/RLP performance at different radio link data rates and a variety of radio channel conditions are shown in Section IV.

III. Effects of Persistence of RLP Frame Error Recovery

In the non-transparent mode, IS-707 uses a NAK (negative acknowledegment) selective repeat ARQ protocol to retransmit lost data frames. The receiver does not acknowledge correct RLP data frames. In case of a data frame loss, RLP performs a partial error recovery through a small number of frame retransmissions, and if retransmissions fail, further error recovery is deferred to TCP layer, which ultimately provides complete end-to-end reliability.

RLP frame losses are notified to the sender via NAK frames, which are sent under control of NAK retransmission timers. Fig.1 shows RLP one-cycle retransmissions controlled by NAK retransmission timers. After this cycle, a NAK abort timer is initiated. The NAK abort timer is implemented, and is considered expired, according to the same rules as a NAK retransmission timer. Thus there are two cycles of frame error recovery in the RLP layer

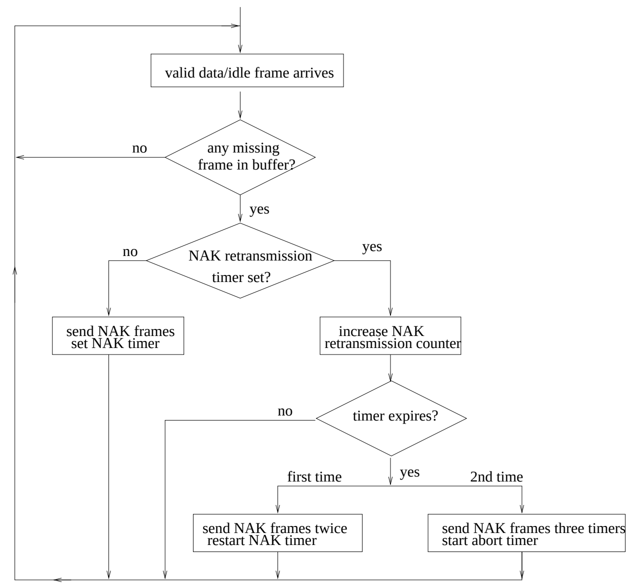

Figure 1: RLP one-cycle retransmissions

controlled by the retransmission timer and abort timer, respectively.

The overall performance improvement by longer persistence at RLP frame error recovery was first shown in [7], where TCP over IS-99 RLP for circuit mode data services was studied for one–hop mobile–base station link, and perfect feedback channel (error-free NAK transmission). The longer persistence was accomplished by simply increasing retransmission attempts.

In this paper, we investigate the effects of longer persistence performance of TCP over IS-707 for packet data services in a wireless Internet access scenario. The longer persistence is accomplished by increasing RLP retransmission cycles. We show that for best-effort bulk data applications like FTP, persisting at the RLP layer and subsequently hiding more frame losses to the TCP layer result in better performance. The performance improvement by increasing RLP retransmission cycles at high FER (frame error rate) is shown in Section IV.

IV. Simulation Results

The simulation model is shown in Fig. 2. The model has been implemented in SSF [9] and executed using Cooperating Systems Corp. C++ implementation of SSF. In the model, the `Mobile Host` (MH) is the receiver of a continuous TCP data flow generated by the `Data Source` at `Remote Host` (RH). We use a TCP Reno variant ported to SSF from the `ns-2` simulator [10]. The TCP maximum segment size (MSS) is set to 536 bytes. Each TCP packet from `TCP Src` is delivered for awnd between 2 and 128. With awnd =1, the performance is degraded due to no sufficient TCP packets available for delivery at low FER regime.

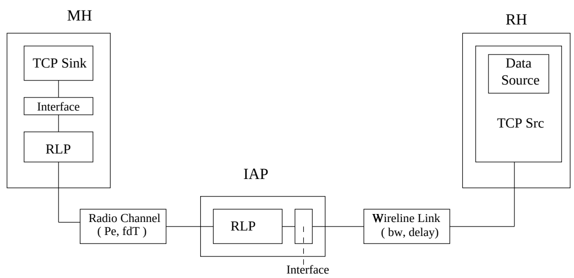

Figure 2: The simulation model.

to the `Interface` at `Internet Access Point` (IAP) and fragmented into IS-707 RLP frames for delivery. At the `TCP Sink` side, the `Interface` reassembles the incoming frames passed from the RLP. The packet size of TCP ACKs from the `TCP Sink` is 40 bytes. `Radio Channel` is modeled as a first-order binary Markov process [11]. By choosing different values of frame error rate P_e and normalized Doppler frequency f_dT, where f_d is the Doppler frequency and T is the frame length (20 ms for IS-707), we can model fading radio channels with different degree of correlation in the fading process. When f_dT is small, the fading process has long-time correlations (long bursts of frame errors); while for large values of f_dT the successive samples of the radio channel are approximately independent. We essentially ignore the wired transmission details in the `Wireline Link` entity, by choosing the bandwidth of 1.5 Mbs and delay of 200ms.

We define the *average TCP goodput* as the average TCP data throughput normalized to the maximum net link throughput; i.e., the average fraction of maximum TCP payload delivery rate. The goodput is upper–bounded by $1 - P_e$.

Long–time average TCP goodput at different advertised window (awnd) sizes is shown in Fig. 3 to Fig. 7 as a function of average FER (frame error rate) P_e for different values of normalized Doppler f_dT. The radio link data rate is 9600bps in these figures.

It is seen that lower FER does not always guarantee better TCP goodput if the awnd size is too large, e.g., in Fig. 3, TCP goodput at $P_e = 0.01$ is worse than that of $P_e = 0.05$ with $f_dT = 0.01$. As explained above, the retransmitted TCP packet takes a long time to be delivered to the receiver at $P_e = 0.01$; On the other hand, at $P_e = 0.05$, due to more frequent TCP packet losses and subsequent smaller congestion window size, there are less TCP packets build-up at IAP and the retransmitted TCP packets take shorter time to be delivered to the receiver, leading to better performance. The TCP goodputs are comparable

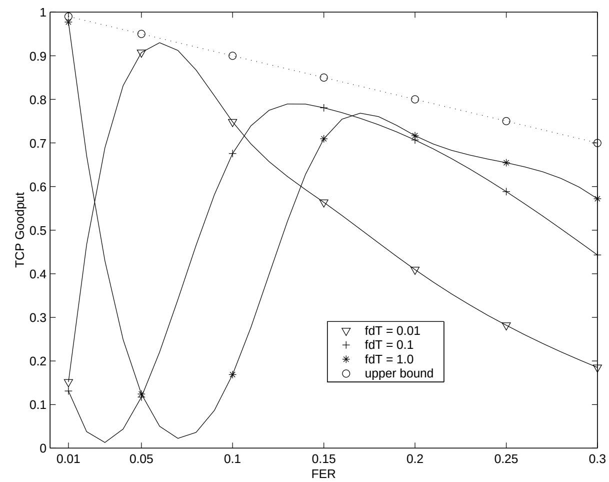

Figure 3: TCP goodput (awnd = 1024, data rate = 9600bps)

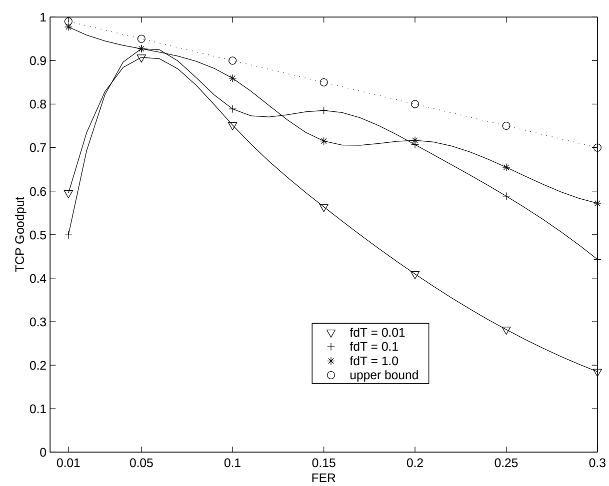

Figure 4: TCP goodput (awnd = 256, data rate = 9600bps)

Next, we investigate the TCP goodput when radio link data rate is 16 $\times$ 9600bps. Fig. 8 to Fig. 10 show TCP goodputs with regard to different awnd sizes. From these figures, it is seen that to achieve desirable performance at this higher link date rate, the advertised window size should be increased above 14.

Finally, we show the effect of longer persistence at the RLP layer to recover lost RLP frames at FER $P_e = 0.3$. It is seen in Fig. 11 that increasing RLP retransmission cycles result in better performance up to a certain value, and the achievable value is easier to obtain for the weakly

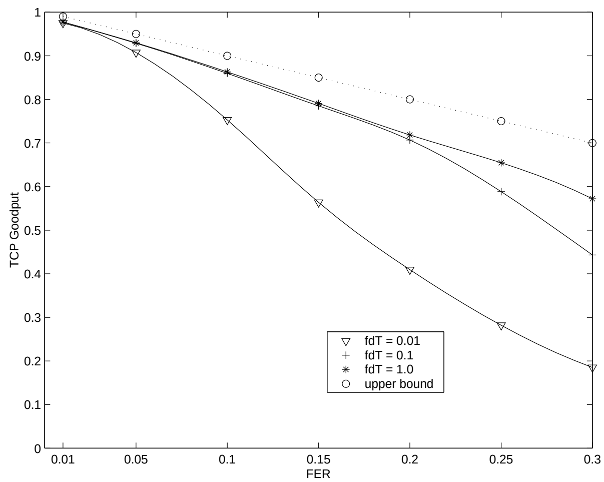

Figure 5: TCP goodput (awnd = 128, data rate = 9600bps)

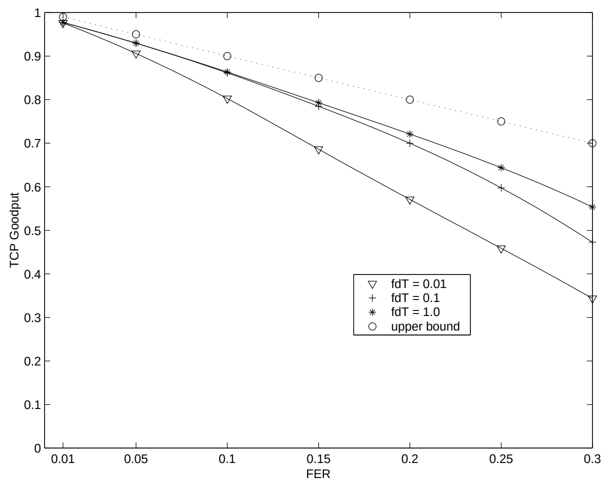

Figure 6: TCP goodput (awnd = 2, data rate = 9600bps)

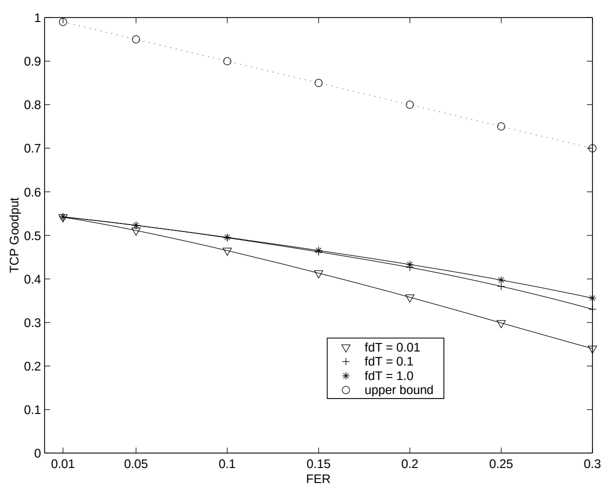

Figure 7: TCP goodput (awnd = 1, data rate = 9600bps)

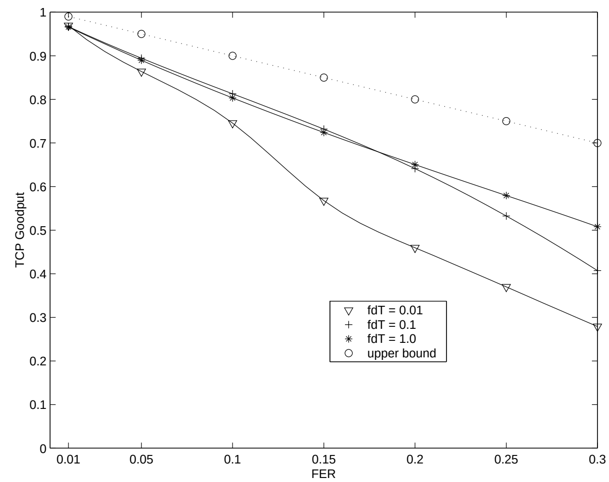

Figure 8: TCP goodput (awnd = 1024, data rate = 16 $\times$ 9600bps)

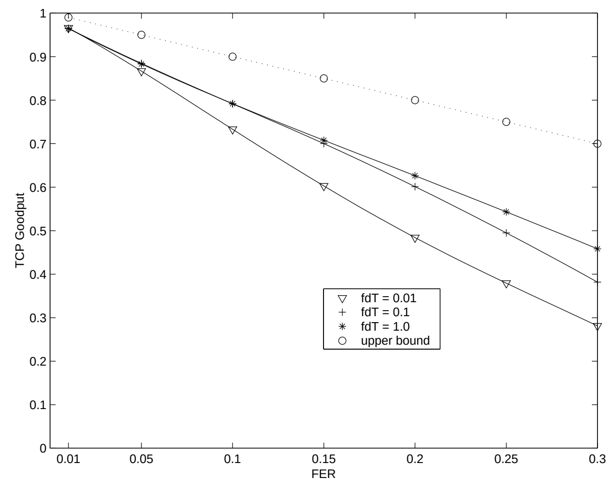

Figure 9: TCP goodput (awnd = 14, data rate = 16 $\times$ 9600bps)

correlated channel conditions ($f_d T = 0.1$ &1.0) as opposed to strongly correlated channel condition ($f_d T = 0.01$). Note that the beneficial effect is obtained for best-effort bulk data applications like FTP. Fig. 12 shows histograms of TCP RTOs (round-trip timeouts) at $P_e = 0.3$, and $f_d T = 1.0$, and varying retransmission cycles with 1, 2, and 10, respectively. It is seen that increase of RLP retransmsion cycles leads to TCP RTO more stable at the expence of a larger mean caused by longer retransmssion latency in the RLP layer.

V. Conclusions

In this paper, the performance and interactions of TCP Reno and IS-707 radio link ARQ protocol in a wireless

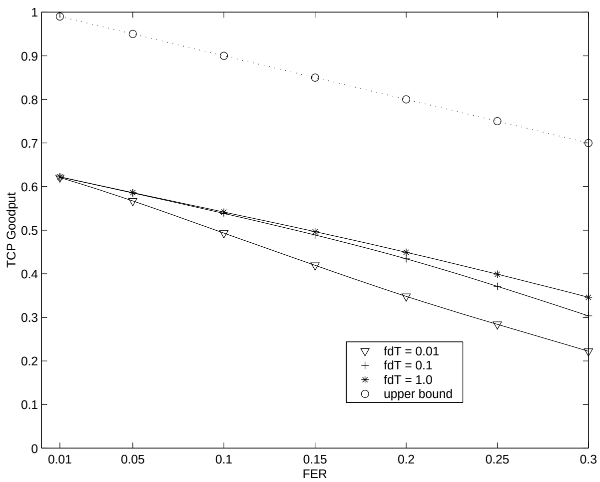

Figure 10: TCP goodput (awnd = 8, data rate = 16 $\times$ 9600bps)

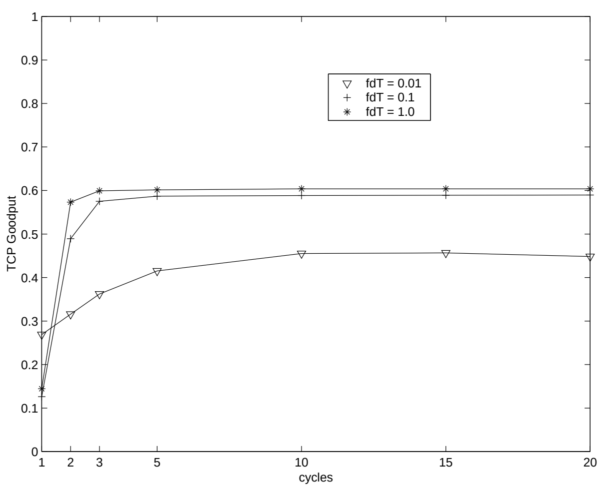

Figure 11: TCP goodput versus RLP retransmission cycles ($P_e = 0.3$)

Internet access scenario was investigated by computer simulation. We demonstrated that an appropriate advertised window size is essential to the overall system performance. We showed that longer persistence in radio link frame error recovery results in better performance for best-effort bulk data applications.

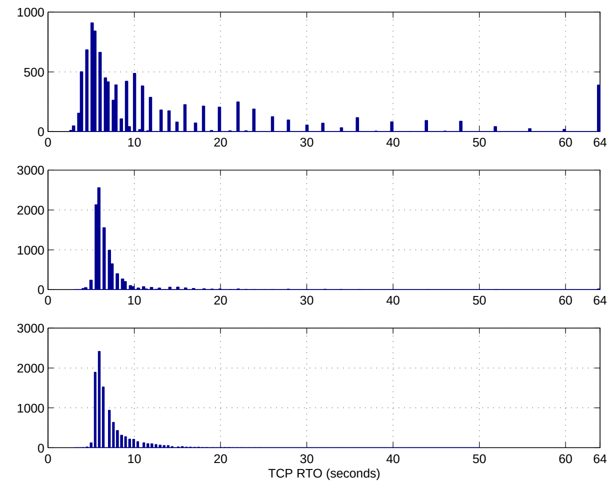

Figure 12: Histogram of TCP RTOs at 1, 2, and 10 RLP retransmission cycles ($P_e = 0.3$, $f_d T = 1.0$)

References

(1) V. Jacobson, “Congestion avoidance and control,” *Computer Communication Review*, vol. 18, no. 4, pp. 314–329, 1988.

(2) P. Karn and C. Partridge, “Improving round-trip time estimates in reliable transport protocols,” in *Proc. of ACM SIGCOMM ’87*, pp. 2–7, Aug. 1987.

(3) TIA/EIA/IS–707, “Data services option standard for wideband spread spectrum digital cellular system,” Feb. 1998.

(4) TIA/EIA/IS–99, “Data services option standard for wideband spread spectrum digital cellular system,” July 1995.

(5) Y. Bai, G. Wu, and A.T. Ogielski, “TCP/RLP Coordination and Interprotocol Signaling for Wireless Internet,” in *Proc. of VTC’99 Spring*, pp. 1945–1951, May 1999.

(6) Y. Bai, A.T. Ogielski, and G. Wu, “TCP over IS-707,” in *Proc. of PIMRC’99 (to appear)*, Sept. 1999.

(7) A. Chockalingam and G. Bao, “Performance of TCP/RLP Protocol Stack on the Correlated Fading DS-CDMA Wireless Links,” in *Proc. VTC’98*, (Ottawa, Canada), pp. 363–367, May 1998.

(8) W. Stevens, *TCP/IP Illustrated Volume I*. Addison-Wesley, 1994.

(9) J. Cowie, D.M. Nicol, and A.T. Ogielski, “Modeling the Global Internet,” *Computing in Science & Engineering*, vol. 1, pp. 42–50, Jan./Feb. 1999.

(10) S. Bajaj et al., “Virtual InterNetwork Testbed: Status and research agenda,” Tech. Rep. 98-678, University of Southern California, July 1998.

(11) M. Zorzi, R. R. Rao, and L. B. Milstein, “On the accuracy of a first-order Markov model for data block transmission on fading channels,” in *Proc. IEEE ICUPC’95*, pp. 211–215, Nov. 1995.